PATENT ABSTRACTS OF JAPAN

(11)Publication number : 09-128349
(43)Date of publication of application : 16.05.1997

(51)Int.Cl. G06F 15/16
G06F 15/16
G06F 11/18

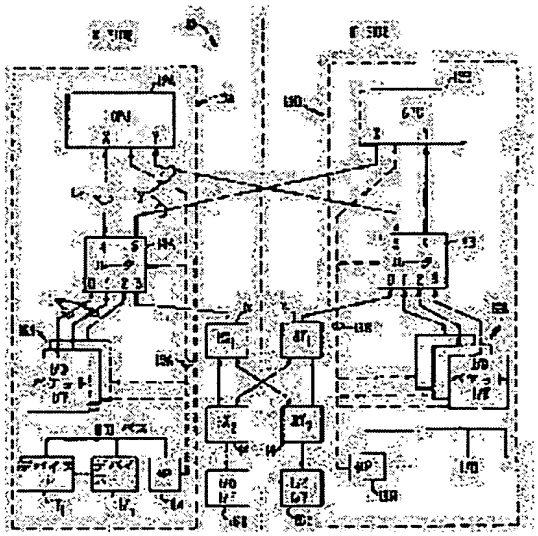
(21)Application number : 08-145550
(22)Date of filing : 07.06.1996

(71)Applicant : TANDEM COMPUT INC
(72)Inventor : ROBERT W HORST
GARCIA DAVID J

(30)Priority
Priority number : 95 485062 Priority date : 07.06.1995 Priority country : US

(54) FAIL-FIRST, FAIL-FUNCTIONAL, AND FAULT-TOLERANT-MULTIPROCESSOR-SYSTEM

(57)Abstract:
PROBLEM TO BE SOLVED: To facilitate fault-tolerant operation by providing a network which mutually connects a central processor and an input/output device so that one of central processors gains communication access to one of input/output devices without requesting other's use.
SOLUTION: The MPs 18 of subprocessor systems 10A and 10B connect an IEEE1149. one-test bus 17 registers used by the MPs 18 to elements of the subprocessor systems thorugh on-line access port interfaces included in the elements so as to transmit states and control information between the elements and MPs 18. The MPs 18 generate and send message packets to communicate with a CPU 12. The CPU 12, a router 14, and an I/O packet interface 16 are mutually connected by a TNet link L and have a two-way data communication.



LEGAL STATUS

[Date of request for examination] 06.06.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)**公開特許公報**（Ａ）　(11)特許出願公開番号

特開平9－128349

(43)公開日　平成９年(1997)５月16日

(51)Int.Cl.[6]		識別記号	庁内整理番号	ＦＩ			技術表示箇所
Ｇ０６Ｆ	15/16	３６０		Ｇ０６Ｆ	15/16	３６０Ｒ	
		４７０				４７０Ｊ	
	11/18	３１０			11/18	３１０Ａ	

審査請求　未請求　請求項の数１　ＯＬ　（全 84 頁）

(21)出願番号　特願平8－145550

(22)出願日　平成８年(1996)６月７日

(31)優先権主張番号　０８／４８５０６２

(32)優先日　1995年６月７日

(33)優先権主張国　米国（ＵＳ）

(71)出願人　391058071
タンデム　コンピューターズ　インコーポレイテッド
TANDEM COMPUTERS INCORPORATED
アメリカ合衆国　カリフォルニア州　95014　クーパーティノ　ノース　タンタウ　アベニュー　10435

(72)発明者　ロバート　ダブリュー　ホースト
アメリカ合衆国　カリフォルニア州　95070　サラトガ　ラーチモント　アベニュー　12386

(74)代理人　弁理士　中村　稔　（外６名）

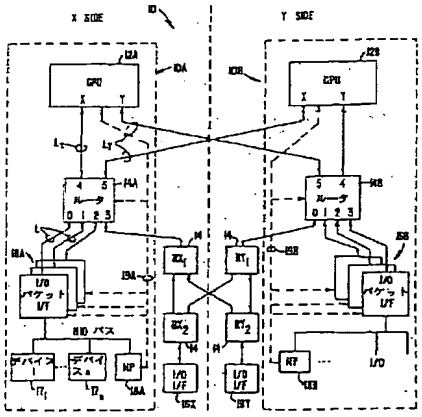
最終頁に続く

(54)【発明の名称】　フェイルーファースト、フェイルーファンクショナル、フォルトトレラント・マルチプロセッサ・システム

(57)【要約】　（修正有）

【課題】　フェイルーファースト、フェイルーファンクショナル、フォルトトレラント・マルチプロセッサ・システムを提供する。

【解決手段】　マルチプロセッサ・システムは、それぞれが実質的に同一に構成された多数のサブープロセッサ・システムを含む。サブープロセッサ・システムの一つのＣＰＵは、システムのＩ／Ｏ装置、またはシステムのＣＰＵと、ルーティング装置を通して、通信しうる。Ｉ／Ｏ装置とＣＰＵｓとの間の通信は、パケット化されたメッセージによって行なわれる。ＣＰＵｓ及びＩ／Ｏ装置は、システムのＣＰＵのメモリへ書込まれるか、またはそれから読取られる。メモリ保護は、そのＣＰＵのメモリへの読取り／書込みに対する妥当性検査を備えているような各ＣＰＵによって保守される。



【特許請求の範囲】
【請求項１】　複数の中央処理装置と、複数の入力／出力装置と、前記複数の中央処理装置のいずれか他のものの使用を要求することなく前記中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの通信アクセスを有するように前記中央処理装置及び前記入力／出力装置を相互接続するネットワークとを備えていることを特徴とする多重処理システム。
【発明の詳細な説明】
【０００１】
【産業上の利用分野】本発明は、一般にデータ処理システムに関し、特に多重処理システム並びにプロセッサ間及び入力／出力連絡の連結性を供給するリライアブル・システム・エリア・ネットワークに関する。更に、システムは、フォルトトレラント機能を示すべく構成される。
【０００２】
【従来の技術】今日のフォルトトレラント・コンピューティングは、特殊な軍事及び通信システムから汎用の高可用性(high availability) 商用(commercial)システムに発展した。フォルトトレラント・コンピュータの発展は、よく記録されている (D. P. Siewiorek, R. S. Swarz, "The Theory and Practice of Reliable System Design", Digital Press, 1982及びA. Avizienis, H. Kopetz, J. C. Laprie, eds, "The Evolution of Fault Tolerant Computing", Vienna: Springer-Verlag, 1987を参照）。初期の高可用性システムは、軍事アプリケーションに対してアイ・ビー・エム (IBM)、ユニヴァック(Univac)、及びレミントン・ランド(Remington Rand)によって1950年代に開発された。1960年代には、アメリカ航空宇宙局(NASA)アイ・ビー・エム(IBM) 、エス・アール・アイ(SRI) 、シー・エス・ドレイパー研究所(C. S. Draper Laboratory) 及びジェット推進研究所(Jet Propulsion Laboratory) は、航空宇宙アプリケーションに対するガイダンス　コンピュータの開発にフォルトトレラントを適用し始めた。また、1960年代には、最初の米国電信電話会社 (ＡＴ＆Ｔ) 製電子スイッチングシステムの開発もあった。
【０００３】最初の商用フォルトトレラント・マシンは、オンライン・トランザクション処理アプリケーション用に1970年代にタンデム・コンピュータによって紹介された(J. Bartlett, "A NonStop Kernal", in proc. Eighth Symposium on OperatingSystem Principles, pp. 22-29, Dec. 1981）。1980年代には、多数他の商用フォルトトレラント・システムが紹介された(O. Serlin, "Fault-Tolerant Systems in Commercial Applications", Computer, pp. 19-30, August, 1984）。現行の商用フォルトトレラント・システムは、分散型メモリ・マルチプロセッサ、共用ーメモリ・トランザクション・ベース・システム、"ペアーアンドースペア(pair-and-spare)" ハードウェア・フォルトトレラント・システム (R. Freiburghouse, "Making Processing Fail-safe", Mini-micro Systems, pp. 255-264, May1982;米国特許第 4,907,228号公報もこのペアーアンドースペア、及び共用ーメモリ・トランザクション・ベース・システム技術の一例である）、及びこの出願及びその中に開示された発明の出願人である、アメリカ合衆国カリフォルニア州カッパティノのタンデム・コンピュータ・インコーポレイティッドによって製造される "インテグリティ(Integrity) " コンピューティング・システムのようなトリプルーモジュラーリダンダント(triple-modular-redundant)・システムを含む。
【０００４】商用フォルトトレラント・コンピュータの殆どのアプリケーションは、オンライン・トランザクション処理に分類される。金融機関は、電子資金移動(electronic funds transfer) 、自動窓口機 (ＡＴＭ) の制御、及び株式市場取引システム(stock market trading systems)に対して高可用性を必要とする。製造メーカーは、自動化された工場の制御、在庫管理、及びオンライン文書アクセス・システムに対してフォルトトレラント・マシンを用いる。フォルトトレラント・マシンの他のアプリケーションは、予約システム、行政データベース、賭け事(wagering)システム、及び電気通信システムを含む。
【０００５】フォルトトレラント・マシンのヴェンダー (Vendors) は、増大したシステム可用性、連続処理の両方、及び故障が存在するときでもデータの正確性を達成することを試みる。特定のシステム・アーキテクチャにより、故障にもかかわらずランを継続するか、または処理が故障が起きたときに最も近いチェックポイントから自動的に再開されるかのいずれかである。あるフォルトトレラント・システムは、故障したコンポーネントについて再構成可能であるべく十分なコンポーネント冗長性を備えているが、故障したモジュールで走行している処理は、失われる。商用フォルトトレラント・システムのヴェンダーは、プロセッサ及びディスクを越えてフォルトトレランスを拡張した。信頼性における大きな改良を行うために、電源、ファン及びモジュール間接続を含んでいる、故障の全てのソース (原因) がアドレスされなければならない。タンデム・コンピュータ・インコーポレイティッドによって製造される "ノン・ストップ(NonStop) " 及び "インテグリティ" アーキテクチャ (この出願の出願人による、両者がそれぞれ広範に説明された米国特許第 4,228,496号公報、米国特許第 5,146,589号公報及び米国特許第 4,965,717号公報；NonStop （ノン・ストップ) 及びIntegrity （インテグリティ) は、タンデム・コンピュータ・インコーポレイティッドの登録商標である）は、商用フォルトトレラント・コンピューティングに対する二つの現行のアプローチを表わす。上記米国特許第 4,278,496号公報に一般に示した、ノン・

ストップ・システムは、一つのハードウェア・コンポーネントの故障にも係わらず動作（オペレーション）を継続すべく設計されたマルチプル・プロセッサ・システムを用いるアーキテクチャを採り入れている。通常動作では、各プロセッサ・システムは、“ホット・バックアップ”としてよりも、その主要コンポーネントを独立にかつ同時に用いる。ノン・ストップ・システム・アーキテクチャは、プロセッサ間連絡に対してバスによって相互接続された１６個のプロセッサ・システムまで含みうる。各プロセッサ・システムは、メッセージ・ベース・オペレーティング・システムのコピーを含むそれ自身のメモリを有する。各プロセッサ・システムは、一つ以上の入力／出力（Ｉ／Ｏ）バスを制御する。Ｉ／Ｏコントローラ及び装置の双方向ポーティング(dual-porting)は、各装置への多重経路を供給する。ディスク記憶装置のような、（プロセッサ・システムに対する）外部記憶装置は、冗長永久データ記憶(redundantpermanent data storage)を維持すべくミラーされうる。

【０００６】このアーキテクチャは、“フェイルーファースト(fail-fast)”動作を供給すべく各システム・モジュールに自己検査ハードウェアを供給する：動作は、他のモジュールの汚染を防ぐためにフォルトが発生したならば一時停止される。フォルトは、例えば、パリティ検査、重合（複写）及び比較、並びに誤り検出コードによって検出される。フォルト検出は、主にハードウェアの責任であり、フォルト回復は、ソフトウェアの責任である。また、ノン・ストップ・マルチープロセッサ・アーキテクチャでは、アプリケーション・ソフトウェア（“プロセス”）は、プライマリ・プロセス及びバックアップ・プロセスを含んでいる、“プロセス－ペア”としてオペレーティング・システム下でシステム上で走りうる。プライマリ・プロセスは、一つの多重プロセッサ上で走り、バックアップ・プロセスは、異なるプロセッサ上で走る。バックアップ・プロセスは、通常、活動停止状態であるが、プライマリ・プロセスからのチェックポイント・メッセージに応じてその状態を周期的に更新する。チェックポイント・メッセージの内容は、完全状態更新の形、または先のチェックポイント・メッセージからの変化だけを連絡する形を取ることができる。最初は、チェックポイントは、アプリケーション・プログラムに手動で挿入されたが、現行では、ほとんどのアプリケーション・コードは、チェックポイントとトランザクション２段階コミット(transaction two-phase commit)プロトコルとの組合せを通して回復を供給するトランザクション処理ソフトウェアの下で走る。

【０００７】タンデム・ノン・ストップ・アーキテクチャのプロセッサ間メッセージ・トラフィックは、各プロセッサが、それ自身を含んでいる、システムの全てのプロセッサによる受信のために“私は、生きている(I'm Alive)”メッセージを周期的にブロードキャストすることを含み、ブロードキャストしているプロセッサがまだ機能していることを他のプロセッサに知らせる。プロセッサが故障したときには、その故障が発表されかつ故障したプロセッサの周期的な“私は、生きている(I'm Alive)”メッセージの欠如によって識別される。応答として、オペレーティング・システムは、最後のチェックポイントからプライマリ・エクスキューション(primary execution)を始めるべく適切なバックアップ処理を導く。新しいバックアップ処理は、別のプロセッサで開始されうるか、または処理は、ハードウェアが修復されるまでバックアップなしで走らせうる。米国特許第 4,817,091号公報は、この技術の一例である。

【０００８】各Ｉ／Ｏコントローラは、それが取り付けられた２つのプロセッサの一つによって管理される。コントローラの管理は、プロセッサ間で周期的に切替えられる。管理しているプロセッサが故障したならば、コントローラの所有権は、他のプロセッサへ自動的に切替えられる。コントローラが故障したならば、データへのアクセスは、別のコントローラを通して維持される。ハードウェア・フォルトトレランスを供給することに加えて、上述したアーキテクチャのプロセッサ・ペアは、ソフトウェア・フォルトトレランスのある測定を供給する。プロセッサがソフトウェア・エラーにより故障するときには、バックアップ・プロセッサは、同じエラーに遭遇することなく処理をしばしば成功裏に継続することができる。バックアップ・プロセッサのソフトウェア環境は、異なるキュー・レングス(queue lengths)、テーブル・サイズ、及びプロセス・ミックスを一般に有する。ソフトウェア品質確認検査を逃れてくるソフトウェア・バグのほとんどが、稀なデータ依存境界条件(infrequent data dependent boundary conditions)を含むので、バックアップ処理は、しばしば成功する。上述したアーキテクチャとは対照的に、インテグリティ・システムは、フォルトトレラント・コンピューティングに対する別のアプローチを示す。１９９０年に紹介された、インテグリティは、ユニックス(Unix)の標準版を走らせるべく設計された（“Unix（ユニックス）”は、アメリカ合衆国デラウエア州のユニックス・システム研究所Inc.の登録商標である）。互換性が主な目的であるシステムでは、ソフトウェアに対する変更をあまり必要としないので、ハードウェア故障回復は、論理的選択である。プロセッサ及び局所メモリは、トリプルーモジュラーリダンダンシー（ＴＭＲ）を用いて構成される。全てのプロセッサは、同じコードストリームを走らせるが、各モジュールの刻時(clocking)は、刻時回路における故障（フォルト）のトレランスを供給すべく独立である。３つのストリームのエクスキューション（実行）は、非同期であり、かついくつかのクロック周期だけ離れてドリフトしうる。ストリームは、グローバル・メモリのアクセスの間中に周期的に再同期される。ＴＭＲコントローラ・

ボード上の有権者(voters)は、プロセッサ・モジュールの故障を検出しかつマスクする。メモリは、三重(triplicated) プロセッサ・ボード上の局所メモリと二重(duplicated) TMRCボード上のグローバル・メモリに区分される。システムの二重部分は、故障を検出するために自己検査技術を用いる。各グローバル・メモリは、双方向ポートされかつプロセッサ並びにI／Oプロセッサ（IOPs）にインターフェイスされる。標準VME周辺コントローラは、バス・インターフェイス・モジュール(Bus Interface Module) (BIM) を通して一対のバスにインターフェイスされる。IOPが故障であるならば、ソフトウェアは、全てのコントローラの制御を残存するIOPに切替えるためにBIMsを用いることができる。ミラー型ディスク記憶装置は、二つの異なるVMEコントローラに取り付けられうる。

【0009】インテグリティ・システムでは、全てのハードウェア故障は、冗長ハードウェアによってマスクされる。修理後、コンポーネントは、オンラインに再統合される。

【0010】

【発明が解決しようとする課題】先の例は、フォルトトレランスをデータ処理システムに組み込んでいる現在のアプローチを示す。ソフトウェア回復を含んでいるアプローチは、少ない冗長ハードウェアを必要とし、かつあるソフトウェア・フォルトトレランスの可能性を与える。ハードウェア・アプローチは、標準オペレーティング・システムを有する完全互換性を許容しかつ他のシステム上で開発されたアプリケーションを透過的に走らせるべくエクストラ・ハードウェア冗長性を用いる。それゆえに、上述したシステムは、ハードウェア（例えば、冗長性を採り入れている、フェイルーファンクショナル）またはソフトウェア技術（例えば、高データ・インテグリティ・ハードウェアを有するソフトウェア回復を採り入れている、フェイルーファースト）のいずれかによりフォルトトレラント・データ処理を供給する。しかしながら、上述したいずれのシステムも、単一データ処理システムによる、ハードウェア（フェイルーファンクショナル）及びソフトウェア（フェイルーファースト）アプローチの両方を用いる、フォルトトレラント・データ処理を供給するように構成されているとは思われない。

【0011】上述したような、コンピューティング・システムは、電子データ交換（EDI）及びグローバル・メッセージングのような、電子商業に対してしばしば用いられる。しかしながら、そのような電子商業における今日の要求は、ユーザの数が増加しかつメッセージが更に複雑になるので、更に多くのスループット能力を要求している。例えば、インターネットの最も広く用いられるファシリティである、テキストーオンリー イー・メール（e－mail）は、毎年かなり成長している。インターネットは、画像、音声、及びビデオ・ファイルを送付するためにますます用いられている。音声記憶及び前送り(voice store-and-forward) メッセージは、いたるところにあるようになり、かつデスクトップ・ビデオ会議及びビデオーメッセージは、ある組織(organization)で受け入れを取得している。メッセージの各型は、更なるスループットを継続的に要求する。そのような環境では、ローカル・エリア・ネットワーク（LANs）等のような種々の通信ネットワークにより相互接続された、並列アーキテクチャが用いられる。

【0012】サーバ・アーキテクチャに対する重要な要求事項は、膨大な量のデータを移動する能力である。サーバは、データボリュームが増加しかつトランザクションが更に複雑になるときに付加されたスループット能力を加えることができるように、スケーラブルな高帯域幅を有すべきである。バス・アーキテクチャは、各システム・コンポーネントに対して利用可能な帯域幅の量を制限する。バス上のコンポーネントの数が増加すると、それぞれに対して更に少ない帯域幅が利用可能である。加えて、即応答は、全てのアプリケーションに対して有利でありかつ対話形アプリケーションに対して必要である。それは、ソース（発生源）からデスティネーション（宛先）にデータを移動するのにどのくらいかかるかの測定である、非常に少ない待ち時間(latency) を必要とする。応答時間に密接に関連した、待ち時間は、サービス・レベル及び被雇用者生産性に影響を及ぼす。本発明の目的は、単一システムで、フォルトトレラント・アーキテクチャ、ハードウェア冗長性及びソフトウェア回復技術に上記二つのアプローチの両方を組み合わせる多重プロセッサ・システムを提供することである。

【0013】

【課題を解決するための手段及びその作用並びに効果】一般に、本発明は、多重サブ（副）処理システムから構成された処理システムを含む。各サブ処理システムは、主要処理素子として、同時に命令ストリームの各命令を実行すべくロックーステップ、同期方式で動作している一対のプロセッサを含む中央処理装置（CPU）を有する。サブ処理システムのそれぞれは、CPU及びサブ処理システムのアソート(assorted)された周辺装置（例えば、マス記憶装置、プリンタ等）を含んでいる、より大きな処理システムの種々のコンポーネント間、並びにより大きな総括処理システムを構成しうるサブープロセッサ間に冗長通信経路を供給する入力／出力（I／O）システム・エリア・ネットワーク・システムを更に含む。処理システムのコンポーネント間（例えば、それがどのサブ処理システムに属しうるかに関係なく、CPUと別のCPU、またはCPUといずれかの周辺装置）の通信は、複数の相互接続しているリンクのバス構造（ここでは、“TNet”と称する）によって相互接続される多数のルータ素子を含んでいるシステム・エリア・ネット

ワーク構造により送信またはソース・コンポーネント（例えば、ＣＰＵ）からデスティネーション（宛先）素子（例えば、周辺装置）に送られるパケット化されたメッセージを形成しかつ送信することによって実施される。ルータ素子は、メッセージ・パケットに含まれた情報に基づいて処理システムの送信コンポーネントから宛先コンポーネントまで適切または利用可能な通信経路を選択する役割を果す。それゆえに、ルータ素子のルーティング能力（機能）は、周辺装置への通信経路をＣＰＵｓのＩ／Ｏシステムに供給するが、プロセッサ間連絡に用いられることをそれに許容もする。

【００１４】上記したように、本発明の処理システムは、“フェイルーファースト”及び“フェイルーファンクショナル”動作の両方を通してフォルトトレラント動作を供給すべく構成される。フェイルーファースト動作は、誤り検査機能をシステムの戦略地点に位置決めすることによって達成される。例えば、各ＣＰＵは、ＣＰＵの（ロックーステップ動作型）プロセッサ素子とその関連メモリとの間の種々のデータ経路における多様な地点で誤り検査機能を有する。特に、本発明の処理システムは、インターフェイスで、かつ性能にほとんど影響を及ぼさないように、誤り検査を実行する。従来技術のシステムは、プロセッサのペアを走らせて、かつデータとプロセッサ及びキャッシュメモリ間の命令フローとを検査（比較）することによって誤り検査を一般に実施する。この誤り検査の技術は、アクセスに遅れを付加する傾向があった。また、この型の誤り検査は、利用可能でありうるオフザシェルフ(off-the-shelf) 部分（即ち、単一半導体チップまたはモジュール上のプロセッサ／キャッシュメモリ組合せ）の使用を排除した。本発明は、主メモリ及びプロセッサーキャッシュ・インターフェイスよりも遅いスピードで動作するＩ／Ｏインターフェイスのような、より遅い速度で動作する地点でプロセッサの誤り検査を実行する。更に、誤り検査は、プロセッサ、それらのキャッシュメモリ、Ｉ／Ｏ及びメモリ・インターフェイスで発生しうる誤りの検出を許容する位置で実行される。これは、それらがパリティまたは他のデータ・インテグリティ検査を必要としないのでメモリ及びＩ／Ｏインターフェイスに対するより簡単な設計を許容する。

【００１５】処理システムのコンポーネント間の通信フローの誤り検査は、システムの素子間に送られるメッセージ・パケットに巡回冗長検査（ＣＲＣ）を付加することによって達成される。各メッセージ・パケットのＣＲＣは、メッセージの宛先においてだけでなく、そのソース（発生源）からデスティネーション（宛先）にメッセージ・パケットを送るために用いられる各ルータ素子により宛先への途中でも検査される。メッセージ・パケットが間違ったＣＲＣを有すべくルータ素子によって見出されたならば、メッセージ・パケットは、そのようにタグ付けされ、かつ保守診断システムに報告される。この特徴は、故障分離に対する有用なツールを供給する。このようなＣＲＣの使用は、メッセージ・パケットが通過するときにルータ素子がＣＲＣを変更または再生しないので終端間でメッセージ・パケットを保護すべく動作する。各メッセージ・パケットのＣＲＣは、各ルータ・クロッシング(router crossing) で検査される。指令記号ー“このパケットは、よい(This packet Good)”（ＴＰＧ）または“このパケットは、悪い(This packet Bad)”（ＴＰＢ）ーは、全てのパケットに添えられる（付加される）。保守診断プロセッサは、誤りが一時（過渡）的であっても、誤りを導くリンクまたはルータ素子を分離するためにこの情報を用いることができる。

【００１６】ルータ素子は、メッセージを受信しかつ送信することができる複数の双方向ポートを備えている。そのように、それらは、多様なトポロジーの用途に適していおり、代替経路を処理システムのいずれかの二つの素子間（例えば、ＣＰＵとＩ／Ｏ装置間）に設けることができ、故障が存在する通信に対して、フォルトトレラント・システムをもたらす。更に、ルータ論理は、メッセージ・パケットが受信されるルータ・ポート及びメッセージ・パケットの宛先に基づいて、ある一定のポートを出力としての考慮からディスエーブルする(disablin g) 機能を含む。そのメッセージ・パケットのルータの出力ポートとしての無許可ポートを示す宛先アドレスを含んでいるメッセージ・パケットを受信するルータは、メッセージ・パケットを捨て、かつ保守診断システムに知らせる。この特徴の適切な使用は、メッセージ・パケットを連続ループ及び遅延に入ることから防ぐかまたは（例えば、以下に説明する、“デッドロック”条件を生成することによって、）そのようにすることから他のメッセージ・パケットを防ぐ。

【００１７】処理システムのＣＰＵｓは、二つの基本モードの一つで動作するように構成される：（ペアの）各ＣＰＵが他とは独立に動作する“シンプレックス・モード”か、または、ＣＰＵｓのペアが同期された、ロックーステップ方式で動作する“デュプレックス・モード”である。シンプレックス・モード動作は、誤り検査ハードウェア（例えば、各プロセッサが、そのシブリング（同胞）プロセッサの動作可能性を検査し、かつ故障したと思われるかまたは信じられるプロセッサの処理を取って代わる機能を有するようなマルチ処理システムを教示する米国特許第4,228,496号公報）によって検出される故障から回復する機能を供給する。デュプレックス・モードで動作するときには、ペアになったＣＰＵｓは、両方ともに同様な命令ストリームを実行し、ペアの各ＣＰＵは、実質的に同時にストリームの各命令を実行する。

【００１８】デュプレックス・モード動作は、あまり堅牢でない(less robust) オペレーティング・システム

（例えば、ユニックス（ＵＮＩＸ）オペレーティング・システム）に対するフォルトトレラント・プラットフォームを供給する。ペアになった、ロックーステップＣＰＵｓを有する、本発明の処理システムは、故障が、主にハードウェアを通して、多くの場合にマスクされる（即ち、故障の存在にも係わらず動作する）ように構成される。処理システムがデュプレックス・モードで動作しているときには、各ＣＰＵペアは、周辺装置が表面上どの（二つ以上の）サブープロセッサ・システムのメンバーであるかに係わりなく、処理システムの周辺装置をアクセスするためにＩ／Ｏシステムを用いる。また、デュプレックス・モードでは、ＣＰＵペアへの送付向けメッセージ・パケットは、ＣＰＵペアの同期、ロックーステップ動作を維持するために実質的に同時にＩ／Ｏシステムによりペアの両方のＣＰＵｓに送付される。それゆえに、本発明の主な創作的態様は、ロックーステップ・ペアの両方のＣＰＵｓが同じ方法で同じ時間にＩ／Ｏメッセージ・パケットを受信することを確実にする機能を有するデュプレックスの動作モードを供給する。これに関しては、デュプレックス・ペアの一つのＣＰＵに接続されたルータ素子は、ペアの両方のＣＰＵ素子に接続される。（マス記憶装置のような周辺装置かまたは処理装置から）ＣＰＵペアに対してメッセージを受信することによって、そのように接続された、ルータは、メッセージを複写しかつＣＰＵｓが同期されたままであることを確実にする同期方法を用いてそれをペアの両方のＣＰＵｓに送付する。結果として、Ｉ／Ｏシステム及び他のデュプレックスＣＰＵペアから視た、デュプレックスＣＰＵペアは、単一ＣＰＵとして見られる。それゆえに、全てのサブ処理システムからの素子を含む、Ｉ／Ｏシステムは、周辺装置がアクセス可能であるような一つのホモジェニアス(homogeneous) システムとしてデュプレックスＣＰＵペアによって見られるようにされる。

【００１９】本発明の別の重要かつ新規な特徴は、ルータ素子の多用（融通）性がいずれかのＣＰＵが実際には一対の同期された、ロックーステップＣＰＵｓであるようなマルチプロセッサ・システムを形成すべくデュプレックス・モード・オペレーティング・システム・ペアのクラスタを結合させるということである。本発明の更に別の重要な態様は、Ｉ／Ｏ素子から発行されている割込みが、他の情報転送と同じ方法で、即ち、メッセージ・パケットによって、ＣＰＵ（またはデュプレックス・モードの場合にはＣＰＵペア）に伝達されるということである。これは、多数の利点を有する： 割込みは、通常のＩ／Ｏメッセージ・パケットのように、ＣＲＣによって保護することができる。また、両方のＣＰＵｓへの同時送付に対する信号への割込み専用の追加信号回線の要求が不要になる；メッセージ・パケット・システムを介して割込みを送付することは、それらが、Ｉ／Ｏメッセージ・パケットと同じ方法で、同期されたファッションでデュプレックスされたＣＰＵｓに到着することを確実にする。割込みメッセージ・パケットは、割込みの原因となる情報を含み、現在行われるように、ＣＰＵ（ｓ）が原因を決定すべく割込みを発行している装置を読取るという時間のかかる要求を不要にする。更に、上記したように、ルーティング素子は、割込みパケット送付に対する多重経路を供給することができ、それによってシステムのフォルトトレラント能力を上昇させる。加えて、Ｉ／Ｏ装置とＣＰＵｓの間でデータを伝達しかつＣＰＵｓに割込みを伝達するために同じメッセージング・システムを用いることは、Ｉ／Ｏ及び割込みの順番を維持する； 即ち、Ｉ／Ｏ装置は、割込みメッセージが送られる前にＩ／Ｏが終了するまで待つ。

【００２０】本発明の更なる新規な態様は、ＣＰＵのメモリへのアクセスの妥当性を検査する技術の実施である。本発明により構成されたような、処理システムは、ＣＰＵのメモリをシステムの他の素子（即ち、他のＣＰＵｓ及び周辺装置）によってアクセスさせる。そうであるならば、不注意な及び／又は無許可なアクセスに対して保護する方法が供給されなければならない。本発明のこの態様に従って、各ＣＰＵは、そのＣＰＵのメモリへのアクセスが許可されるＣＰＵの外部の各ソースに対するエントリを含んでいるアクセス妥当性検査及び変換（ＡＶＴ）表を維持する。各そのようなＡＶＴ表エントリは、許容されたアクセスの型（例えば、メモリへの書込み）、及びそのアクセスが許容されるメモリの場所に対するような情報を含む。Ｉ／Ｏシステムを通して送られるメッセージ・パケットは、メッセージ・パケットのオリジネータ(originator)、メッセージ・パケットの宛先、メッセージが含んでいるもの（例えば、宛先で書込まれるべきデータ、または宛先から読取られるべきデータに対する要求）、等を記述している情報を有して、上述したように、生成される。ルータ素子にその最終宛先にメッセージ・パケットを迅速に送らせることに加えて、受信ＣＰＵは、メッセージ・パケットのソースに関係している（属している）エントリに対してＡＶＴ表をアクセスし、アクセスが許容されるかどうか、かつそうであるならば、受信ＣＰＵがリマップ（即ち、変換）すべく選択するアドレスの型及びその場所を理解するために検査すべく情報を用いる。この方法で、ＣＰＵのメモリは、逸脱したアクセスに対して保護される。また、ＡＶＴ表は、ＣＰＵへ割込みを通過するためにも用いられる。

【００２１】ＡＶＴ表は、ＣＰＵｓのメモリが故障のＩ／Ｏ装置によって汚染されないことを保証する。アクセス権は、１バイトからページの広がりまでの大きさの範囲であるフォーム・メモリ(form memory) に付与することができる。システムのシステム・ベンダーは、サードパーティー周辺サプライヤのハードウェア及びソフトウェアの品質に関してより少ない制御を通常有するので、

この故障封じ込め(fault containment)は、Ｉ／Ｏにおいて特に重要である。問題は、Ｉ／Ｏシステム全体よりも単一のＩ／Ｏ装置またはコントローラに分離することができる。本発明の更なる態様は、データをＩ／Ｏに送信するためにＣＰＵによって用いられる技術を含む。本発明のこの態様によれば、ＣＰＵとプロセッサ・システムの他のコンポーネントとの間の入力／出力情報転送を処理するために各ＣＰＵにブロック転送エンジンが備えられる。それによって、ＣＰＵの個別のプロセッサ装置は、メモリからＴＮｅｔネットワーク上に情報を得ること、またはネットワークから情報を受け入れることのよりありふれたタスク(more mundane tasks)から取り除かれる。ＣＰＵのプロセッサ装置は、所望の宛先、データの量、及び応答が必要であるならば、受信したときに応答が配置されるメモリの場所、のような他の情報を伴った、送られるべきデータを含んでいるメモリにデータ構造を単にセットアップする。プロセッサ装置がデータ構造を生成するタスクを終了したときには、ブロック転送エンジンは、それが取って代わり、かつメッセージ・パケットの形で、データを送ることを始める原因となることが知らされる。応答が期待されるならば、ブロック転送エンジンは、応答が行くメモリの場所を含んでいる、応答を処理するために必要な構造をセットアップする。応答が受信されるとき及び受信されたならば、それは、識別された期待メモリ位置に送られ、かつ応答を受信したことをプロセッサ装置に知らせる。

【００２２】本発明の更なる態様及び特徴は、添付した図面に関連して本発明の以下の詳細な説明を読むことにより当業者に明らかになるであろう。

【００２３】

【実施例】

概要：ここで、図１を参照すると、本発明の種々の教示により構成された、データ処理システムが参照番号１０で示されている。図１が示すように、データ処理システム１０は、それぞれの構造及び機能が実質的に同じである二つのサブプロセッサシステム１０Ａ及び１０Ｂを備えている。従って、特に示さない限り、サブプロセッサシステム１０のいずれ一つの説明は、他のサブプロセッサシステム１０に同様に適用されるということが理解されるべきである。従って、図１を続いて参照すると、サブプロセッサシステム１０Ａ，１０Ｂのそれぞれは、中央処理装置（ＣＰＵ）１２、ルータ１４、及びそれぞれがネイティブ（固有の）入力／出力（ＮＩＯ）バスによって多数（ｎ）のＩ／Ｏ装置１７に結合される複数の入力／出力（Ｉ／Ｏ）パケット・インターフェイス１６を含んで示されている。Ｉ／Ｏパケット・インターフェイス１６の少なくとも一つは、保守プロセッサ（ＭＰ）１８にも結合される。

【００２４】各サブプロセッサシステム１０Ａ，１０ＢのＭＰ１８は、ＩＥＥＥ１１４９．１テスト・バス１７（図１では、想像線で示す；明瞭化のために図２及び図３では示されていない）及び、状態及び制御情報を素子とＭＰ１８の間に伝達（通信）するためにＭＰ１８によって用いられるレジスタを、各素子に対して、含むオンライン・アクセス・ポート（ＯＬＡＰ）インターフェイスを介してそのサブプロセッサシステムの素子のそれぞれに接続する。ＭＰ１８は、メッセージ・パケットを生成しかつ送ることによって、図１に示すように、ＣＰＵｓ１２とも通信することができる。（実際には、ＭＰ１８からの要求に応じてパケットを生成しかつ送るのは、Ｉ／Ｏパケット・インターフェイス１６である。）ＣＰＵ１２、ルータ１４、及びＩ／Ｏパケット・インターフェイス１６は、“ＴＮｅｔ”リンクＬによって相互接続され、双方向データ通信を供給する。各ＴＮｅｔリンクＬは、二つの一方向１０ビット・サブリンク・バスを備えている。各ＴＮｅｔサブリンクは、データの９ビット及び付随クロック信号を運ぶ。図１に更に示すように、ＴＮｅｔリンクＬは、サブプロセッサシステム１０Ａ及び１０Ｂを互いに相互接続もし、各サブプロセッサシステム１０に他の並びにＣＰＵ間通信のＩ／Ｏ装置へのアクセスを供給する。理解されるように、処理システム１０のＣＰＵ１２は、そのようなアクセスは確証されなければならないが、他のＣＰＵ１２のメモリへのアクセスを付与されることができる　－　本発明の重要な側面である。ある程度類似なファッションで、ＣＰＵ１２のメモリは、通常ＣＰＵによって起動された動作の結果として、周辺装置へもアクセス可能である。これらのアクセスは、方向の定まらない(wayward)周辺装置１７によるＣＰＵ１２のメモリの汚染を防ぐためにも確証される。

【００２５】サブプロセッサシステム１０Ａ／１０Ｂは、図１（及び以下に説明する図２、図３）に示すように一対になるのが好ましく、各サブプロセッサシステム１０Ａ／１０Ｂペアは、ＣＰＵ１２、少なくとも一つのルータ１４、及び関連Ｉ／Ｏ装置を有する少なくとも一つのＩ／Ｏパケット・インターフェイス１６を含んでいる。

【００２６】各ＣＰＵ１２は、そこにおいてメッセージ・パケットが送信及び／又は受信される、二つのＩ／Ｏポート、Ｘポート及びＹポートを有する。ＣＰＵ１２（例えばＣＰＵ１２Ａ）のＸポートは、対応サブプロセッサシステム（例えば１０Ａ）のルータ（１４Ａ）へＴＮｅｔリンクＬによって、接続する。逆に、Ｙポートは、ＣＰＵ（１２Ａ）をコンパニオン・サブプロセッサシステム（１０Ｂ）のルータ（１４Ｂ）に接続する。この後者の接続は、他のサブプロセッサシステム（１０Ｂ）のＩ／Ｏ装置へのＣＰＵ（１２Ａ）によるアクセスに対する通信経路だけでなく、ＣＰＵ間通信に対してそのシステムのＣＰＵ（１２Ｂ）への通信経路も供給する。情報は、メッセージ（パケット）を介して処理シス

テム１０の素子とシステムの他の素子（例えば、サブプロセッサシステム１０AのＣＰＵ１２Ａ）及びシステムの他の素子（例えば、サブプロセッサシステム１０Ｂの Ｉ／Ｏパケット・インターフェイス１６Ｂに関連したＩ／Ｏ装置）との間で伝達される。各メッセージ・パケットは、データを含みうるかまたは指令記号でありうる多数の９ビット記号で構成される。メッセージ・パケットは、メッセージ・パケットを送信しているコンポーネントによって供給される送信機クロックを伴った、ビット一並列、記号一直列ファッションで、ＴＮｅｔリンクＬ上に同期的に送信される。通信素子（即ち、送信機及び受信機）間のクロックは、二つのモード、＜近周波数（near frequency）＞モード、または＜周波数封じ込み（frequency locked）＞モードの一つで動作されうる。

【００２７】近周波数で動作するときには、送信素子及び受信素子によって用いられるクロック信号は、所定のトレランス（許容範囲）内で一実質的に同じ周波数であるべく制約されるが、別々であり、かつ局所的に生成される。この理由により、クロック同期先入り先出し（ＣＳ　ＦＩＦＯ）記憶装置構造（以下に詳述）を用いて、受信機で記号を受信する固有の方法が開発された。ＣＳ　ＦＩＦＯは、近周波数動作の結果としてメッセージ・パケットの受信機と送信機のクロック信号間に起こりうるスキューを吸収すべく動作する。近周波数動作は、一つのルータ１４から別のルータへ、またはルータ１４とＩ／Ｏパケット・インターフェイス１６との間で、またはルータ１４とシンプレックス・モード（以下に詳述）で動作しているＣＰＵｓ１２の間で記号を送信するときに用いられる。

【００２８】周波数封じ込み動作は、同相である必要はないが、送信機及び受信機装置のクロック信号の周波数が封じ込め(lock)られることを意味する。周波数封じ込みクロック信号は、ルータ１４Ａ，１４Ｂと対になったサブプロセッサシステム（例えば、図１のサブプロセッサシステム１０Ａ、１０Ｂ）のＣＰＵｓ１２との間で記号を送信するために用いられる。送信及び受信素子のクロックは、位相関係にないので、近周波数動作に対して用いられるものとは多少異なるモードで動作するにもかかわらず一一クロック同期ＦＩＦＯ（ＣＳ　ＦＩＦＯ）が再び用いられる。各ルータ１４は、一つを除き、それぞれが実質的の同じように構成される、６個の双方向ＴＮｅｔポート、０～５を備えている：ＣＰＵ１２に接続するために用いられる二つのポート（４，５）は、ある程度異なって構成される。この相違は、理解されるように、サブプロセッサシステム１０のペアが、各ＣＰＵ１２が同じ命令ストリームから同時に同じ命令を実行するために動作する、デュプレックス・モードと呼ばれる、同期された、ロックーステップ・モードで動作することができるという事実による。デュプレックス・モードでは、ある一つのＩ／Ｏ装置からの入力Ｉ／Ｏは、実質的に同じ時間に両方のＣＰＵｓ１２に供給されるということが重要である。それゆえに、例えば、ルータ１４Ａのポート３で受信したメッセージ・パケットは、ルータ１４Ａによって複写（複製）されかつ同じ記号が実質的に同じ時間にＣＰＵｓ１２へ伝達されるようにルータ４，５から送信される。ポート４，５がルータ１４の他のポート０～３とは異なるのは、この点においてである。

【００２９】図１は、本発明の別の特徴である、追加のルータ１４（図１では、ルータRX_1，RX_2，RY_1，及びRY_2で示される）の使用による二つのサブプロセッサシステム１０Ａ，１０Ｂ間のクロス・リンク接続を示す。図１に示すように、追加されたルータRX_1，RX_2，RY_1，及びRY_2は、それらをＩ／Ｏパケット・インターフェイス１６Ｘ，１６Ｙに結合するためにサブプロセッサ１０Ａ，１０Ｂ（または、示したように、それぞれＸ及びＹ“側”）間にクロス・リンク接続を形成する。ルータRX_1－RY_2及びRY_1－RX_2間のクロス接続リンクは、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ｂ，１４Ａの間のクロス・リンク接続Ｌｙと同じように一つの側（ＸまたはＹ）から別の側へクロスーリンク経路を供給する。しかしながら、ルータRX_1，RX_2，RY_1，及びRY_2によって供給されるクロス・リンクは、Ｉ／Ｏパケット・インターフェイス１６Ｘ，１６Ｙに接続されうるＩ／Ｏ装置（図示省略）を一つの側（ＸまたはＹ）又は別の側へ送らせる。図１に示すように、ルータRX_2及びRY_2は、Ｉ／Ｏパケット・インターフェイス装置１６ｘ及び１６ｙにデュアル・ポートされたインターフェイスを供給する。もちろん、Ｉ／Ｏパケット・インターフェイス１６Ｘ，１６Ｙが、ルータRX_2及びRY_2によって形成されたデュアル・ポート接続によって供給されたクロスーリンク接続に対する代替としてデュアル・ポート及びルータRX_1，RY_1に接続するためのそれらデュアル・ポートを有すべくそれ自身を構成することができるということは、いま明らかである。

【００３０】ルータ１４の構成及び設計が理解されたときに明らかになるように、それらは、図２及び図３に示すような追加のサブプロセッサシステムを含むべく処理システム１０の構造を拡張できるように用いるのに向いている。図２では、例えば、ルータ１４Ａ及び１４Ｂのそれぞれの一つのポートは、対応サブプロセッサシステム１０Ａ及び１０Ｂを追加のサブプロセッサシステム１０Ａ’及び１０Ｂ’に接続するために用いられ、それにより、図１の基本処理システム１０のクラスタを含んでいる更に大きな処理システムを形成する。同様に、図３において上記の内容は、サブプロセッサシステム・ペア１０Ａ／１０Ｂ，１０Ａ’／１０Ｂ’，１０Ａ’’／１０Ｂ’’，及び１０Ａ’’’／１０Ｂ’’’を備えている、８つのサブプロセッサシステム・クラスタを形成すべく拡張される。次に、サブプロセッサシステムのそれ

ぞれ（例えば、サブプロセッサシステム１０Ａ）は、図３が示すように、サブプロセッサシステム１０Ａ及び１０Ｂが、サブプロセッサシステム１０Ａ’／１０Ｂ’，１０Ａ’’／１０Ｂ’’，及び１０Ａ’’’／１０Ｂ’’’を越えてクラスタを拡張するために、追加のルータ１４Ｃ及び１４Ｄをそれぞれ含むということを除き、ＣＰＵ１２、ルータ１４、及びＩ／Ｏパケット・インターフェイス１６によってＴＮｅｔに接続されたＩ／Ｏの同じ基本最小構成を本質的に有する。更に図３に示すように、ルータ１４Ｃ及び１４Ｄの未使用ポート４及び５は、クラスタを更に拡張するために用いうる。

【００３１】ルータ１４の設計、並びにシステム１０のトポロジーを構成するときのルータ１４の適切な使用と共に、メッセージ・パケットを送るために用いる方法により、図３の処理システム１０のＣＰＵ１２は、他のサブプロセッサシステムの他の“端末装置”（例えば、ＣＰＵ又は／及びＩ／Ｏ装置）をアクセスできる。二つの経路が、ＣＰＵ１２からＩ／Ｏパケット・インターフェイス１６に接続している最後のルータ１４まで利用可能である。例えば、サブプロセッサシステム１０Ｂ’のＣＰＵ１２Ｂは、（サブプロセッサシステム１０Ｂ’）のルータ１４Ｂ、ルータ１４Ｄ、及び（サブプロセッサシステム１０Ｂ’’’）のルータ１４Ｂ及び、リンクＬＡを介して、ルータ１４Ａ（サブシステム１０Ａ’’’）、（サブシステム１０Ａ’の）ルータ１４Ａを介するＯＲ、ルータ１４Ｃ、及びルータ１４Ａ（サブシステム１０Ａ’’’）を介してサブプロセッサシステム１０Ａ’’’のＩ／Ｏ１６’’’をアクセスできる。同様に、サブ処理システム１０Ａ’’のＣＰＵ１２Ａは、データを読取りまたは書込むためにサブプロセッサ１０ＢのＣＰＵ１２Ｂに含まれるメモリを（二つの経路を介して）アクセスしうる。（処理システムの別のコンポーネントの一つのＣＰＵ１２によりメモリ・アクセスは、分かるように、そのようにするための許可を有することのアクセスを求めているコンポーネントを必要とする。この点において、各ＣＰＵ１２は、そのＣＰＵのメモリをアクセスするための許可を有している各コンポーネントに対するエントリを含み、通常、そのアクセスをメモリの選択されたセクションに制限し、かつ許可したアクセスの型を制限している、表を維持する。このような方法で許可を要求することは、誤ったアクセスによるＣＰＵのメモリ・データの汚染を防ぐ。）

【００３２】図２に示す処理システムのトポロジーは、サブプロセッサシステム１０Ａ’，１０Ｂ’のルータ１４Ａ’，１４Ｂ’に接続すべく、ルータ１４Ａ、１４Ｂのポート１、及び補助ＴＮｅｔリンクＬＡを用いて達成される。それによって得られたトポロジーは、図２に示す処理システム１０のＣＰＵ１２（１２Ａ，１２Ｂ，１２Ａ’，１２Ｂ’）とＩ／Ｏパケット・インターフェイス１６との間に冗長通信経路を確立する。例えば、サブプロセッサシステム１０Ａ’のＣＰＵ１２Ａ’は、ルータ１４Ａ’（イン　ポート４、アウト　ポート３）、ルータ１４Ａ（イン　ポート３、アウト　ポート０）、及び関連相互接続ＴＮｅｔリンクＬによって形成された第１の経路によりサブプロセッサシステム１０ＡのＩ／Ｏ１６Ａをアクセスしうる。しかしながら、ルータ１４Ａ’が失われたならば、ＣＰＵ１２Ａ’は、ルータ１４Ｂ’（イン　ポート４、アウト　ポート３）、ルータ１４Ｂ（イン　ポート３、アウト　ポート１）、リンクＬＡ、及びルータ１４Ａ（イン　ポート１、アウト　ポート０）によって形成された経路によりＩ／Ｏ　１６Ａをアクセスしうる。また、図２のトポロジーは、システム１０のＣＰＵｓ１２のペア間に冗長通信経路をも確立し、フォルトトレラントＣＰＵ間通信に対する手段を供給するということに注目する。

【００３３】図３は、図２に示したもののトポロジーの拡張を示す。各サブプロセッサ・ペアの各ルータ１４の一つのポートを相互接続し、サブプロセッサシステム１０Ａ’’，１０Ｂ’’及び１０Ａ’’’，１０Ｂ’’’のルータ１４（１４Ａ’’及び１４Ｂ’’）のポート１間に追加の補助ＴＮｅｔリンクＬＡ（鎖線接続で図３に示す）を用いることによって、二つの分離、独立データ経路がＣＰＵ１２とＩ／Ｏパケット・インターフェイス１６との間に見出すことができる。このファッションでは、端末装置（例えば、ＣＰＵ１２またはＩ／Ｏパケット・インターフェイス１６）は、他の端末装置への少なくとも二つの経路を有する。二つの端末装置間（例えば、図３のシステム１０における、ＣＰＵ１２と他のＣＰＵ１２の間、またはＣＰＵ１２とＩ／Ｏパケット・インターフェイス１６との間）のアクセスの代替経路を供給することは、重要な概念である。フォルト・ドメイン(fault domain)の損失は、二つの残っているフォルト・ドメイン間の通信を分断しない。ここで、フォルト・ドメインは、サブプロセッサシステム（例えば、１０Ａ）でありうる。それゆえに、ルータ１４Ａ’’’及び１４Ｂ’’’間に補助ＴＮｅｔリンクＬＡなしで、電力が供給される故障によりサブプロセッサシステム１０Ａが破壊されたならば、サブプロセッサシステム１０ＢのＣＰＵ１２Ｂは、（ルータ１４Ａ、ルータ１４Ｃ、ルータ１４Ａ’’’を介して、Ｉ／Ｏパケット・インターフェイス１６’’’へ）Ｉ／Ｏパケット・インターフェイス１６’’’へのアクセスを失うであろう。ルータ１４Ａ’’’及び１４Ｂ’’’間の補助接続ＬＡで、サブプロセッサシステム１０Ａの損失のよるルータ１４Ａ（及びルータ１４Ｃ）の損失を伴っても、ＣＰＵ１２Ｂ間の通信は、ルータ１４Ｂのルート、ルータ１４Ｄ、ルータ１４Ｂ’’’、ルータ１４Ａ’’’へ、及び最後にＩ／Ｏパケット・インターフェイス１６’’’への補助接続ＬＡを介してまだ可能である。

【００３４】ＣＰＵアーキテクチャ：ここで図４を参照

すると、ＣＰＵ１２Ａがより詳細に示されている。ＣＰＵ１２Ａ及び１２Ｂの両方は、構成及び機能が実質的に同じなので、ＣＰＵ１２Ａの詳細だけを説明する。しかしながら、特に示さない限り、ＣＰＵ１２Ａの説明がＣＰＵ１２Ｂにも同様に適用するということが理解されるであろう。図４に示すように、ＣＰＵ１２Ａは、両方のプロセッサ装置２０ａ，２０ｂが同じ命令を受信しかつ実行し、そして時間において実質的に同じ瞬間に、同じデータ及び指令出力を発行するように、同期した、ロックーステップ動作に対して構成される一対のプロセッサ装置２０ａ，２０ｂを含む。プロセッサ装置２０ａ，２０ｂのそれぞれは、対応キャッシュメモリ２２に、バス２１（２１ａ，２１ｂ）によって、接続される。用いられる特定の型のプロセッサ装置は、キャッシュメモリ２２が必要でないように十分な内部キャッシュメモリを含むことができる。代替的に、キャッシュメモリ２２は、プロセッサ装置２０の内部でありうるキャッシュメモリを補うために用いることができる。とにかく、キャッシュメモリ２２が用いられたならば、バス２１は、１２８ビットのデータ、１６ビットの誤り訂正コード（ＥＣＣ）チェックビットを導通すべく構成され、データ、（データ及び対応ＥＣＣに対する）２５のタグ・ビット、タグ・ビットをカバーしている３つのチェックビット、２２のアドレス・ビット、アドレスをカバーしている３ビットのパリティ、及び７つの制御ビットを保護する。

【００３５】また、プロセッサ２０ａ，２０ｂは、Ｘ及びＹインターフェイス装置２４ａ，２４ｂに分離６４ビット・アドレス／データ・バス２３を介して、それぞれ結合される。所望ならば、各バス２３ａ，２３ｂ上で伝達されたアドレス／データは、これはバスの幅を増大するが、パリティによっても保護することができる。（プロセッサ２０は、アメリカ合衆国カリフォルニア州サンタ・クララのシリコン・グラフィックス，インク(Silicon Graphics, Inc.)のＭＩＰＳディビジョンから入手可能なような、ＲＩＳＣ　Ｒ４０００型マイクロプロセッサを含むべく構成されるのが好ましい。）

Ｘ及びＹインターフェイス装置２４ａ，２４ｂは、プロセッサ装置２０ａ，２０ｂと（二つのＭＣハーフ２６ａ及び２６ｂからなる）メモリ・コントローラ（ＭＣ）２６及びダイナミック・ランダム・アクセス・メモリ・アレー２８を含んでいるＣＰＵ１２Ａのメモリ・システムとの間にデータ及び指令信号を伝達すべく動作する。インターフェイス装置２４は、７２ビット・アドレス／指令バス２５により互いにかつＭｃｓ２６ａ及び２６ｂに相互接続する。しかしながら、見て分かるように、（ＥＣＣの８ビットを伴った）データの６４ビット倍長語（ダブルワード）は、インターフェイス装置２４　によってメモリ２８に書込まれ、一つのインターフェイス装置２４は、書込まれる倍長語の一つの語（例えば、上位３２ビット部分）だけを駆動し、他のインターフェイス装置２４は、倍長語の他の語（例えば、倍長語の下位３２ビット部分）。更に、各書込み動作で、インターフェイス装置２４ａ，２４ｂは、誤りを検査するためにそのインターフェイス装置２４によって書込まれないデータ上で他のものによって書込まれたデータでクロスチェック動作を実行する；また、読取り動作では、バス２５上に置かれたアドレスが同じ方法でクロス・チェックされる。キャッシュメモリ２２並びに（主）メモリ２８に書き込まれたデータの両方を保護するために用いられる特定のＥＣＣは、通常のものであり、かつ単一ビット誤り訂正、ダブルビット誤り検出を供給する。

【００３６】概念的に、各倍長語は、“奇数”及び“偶数”語を含む。Ｍｃｓ２６の一つは、メモリに奇数語を書込み、他のものは、偶数語を書込む。更に、Ｍｃｓ２６は、その倍長語に対する８ビット誤り訂正コード（ＥＣＣ）と一緒に、一度に二つの倍長語を書込む。加えて、ＥＣＣチェックビットは、倍長語だけでなく、倍長語が書込まれるメモリ位置のアドレスをカバーするためにも形成される。後者がアクセスされたときには、ＥＣＣは、単一ビット誤りを修正し、かつアクセスされた倍長語が、倍長語がそれから記憶された位置のアドレスに対応することを検査すると同時にデータに発生しうる、ダブルビット誤りを検出するために用いられる。ＣＰＵ１２Ａのインターフェイス装置２４ａ，２４ｂは、ＣＰＵ１２ＡのＸ及びＹ（Ｉ／Ｏ）ポートをそれぞれサービスするための回路素子を形成する。それゆえに、Ｘインターフェイス装置２４ａは、プロセッサシステム１０Ａ（図１）のルータ１４Ａのポートに双方向ＴＮｅｔリンクＬｘによって接続し、Ｙインターフェイス装置２４ｂは、ＴＮｅｔリンクＬｙによりプロセッサシステム１０Ｂ（図１）のルータ１４Ｂに同様に接続する。Ｘインターフェイス装置２４ａは、ルータ１４Ａとサブプロセッサシステム１０ＡのＣＰＵ１２Ａとの間の全てのＩ／Ｏトラフィックを処理する。同様に、Ｙインターフェイス装置２４ｂは、ＣＰＵ１２Ａとサブプロセッサシステム１０Ｂのルータ１４Ｂとの間の全てのＩ／Ｏトラフィックに対して役割を果たす。

【００３７】Ｘインターフェイス装置２４ａをルータ１４Ａに接続しているＴＮｅｔリンクＬｘ（図１、図２及び図３）は、上述したように、それぞれがクロック信号、及び９ビットのデータを運ぶ、二つの１０ビット・バス30_X，32_Xを含む。バス30_Xは、ルータ１４Ａに送信されたデータを運ぶ；バス32_Xは、ルータ１４Ａから入力してくるデータを運ぶ。同様なファッションで、Ｙインターフェイス装置２４ｂは、一緒にＴＮｅｔリンクＬｙを形成している二つの１０ビット・バス３０（出力送信に対する）$_Y$及び32_Y（入力送信に対する）により（サブプロセッサシステム１０Ｂの）ルータ１４Ｂに接続される。Ｘ及びＹインターフェイス装置２

４ａ，２４ｂは、ロックーステップで同時の動作され、実質的に同じ時間に同じ動作を実質的に実行する。それゆえに、Ｘインターフェイス装置２４ａだけがデータをバス30_x上に実際に送信するが、同じ出力データがＹインターフェイス装置２４ｂによって生成され、そして誤り検査に用いられる。Ｙインターフェイス装置２４ｂ出力データは、それがＸインターフェイス装置２４ａによって受信されかつＸインターフェイス装置によって生成される同じ出力データに対して比較されるところのクロス・リンク34_yによってＸインターフェイス装置２４ａに結合される。このように、ＣＰＵ１２ａのＸポートで利用可能にされる出力データは、誤りについて検査される。

【００３８】同じファッションで、ＣＰＵ１２Ａのポートから送信された出力データが検査される。Ｙインターフェイス装置２４ｂからの出力データは、１０ビット・バス30_yによりＹポートに、かつＸインターフェイス装置によって生成されたもので検査される９ビット・クロス・リンク34_yによりＸインターフェイス装置２４ａに結合される。

【００３９】上述したように、二つのインターフェイス装置２４ａ，２４ｂは、互いに同期の、ロックーステップで動作し、それぞれが同時に同じ動作を実質的に実行する。この理由で、ＣＰＵ１２ＡのＸ及び／又はＹポートで受信したデータは、このロックーステップ・モードに二つのインターフェイス装置を維持すべく両方のインターフェイス装置２４ａ，２４ｂによって受信されなければならない。それゆえに、一つのインターフェイス装置（２４ａまたは２４ｂ）によって受信されたデータは、破線及び（Ｘインターフェイス装置２４ａによってＸポートで受信される入力データをＹインターフェイス装置２４ｂに伝達する）９ビット・クロス・リンク接続36_x及び（Ｙインターフェイス装置２４ｂによってＹポートで受信したデータをＸインターフェイス装置２４ａに伝達する）36_yに示されるように、他のインターフェイス装置（２４ｂまたは２４ａ）に渡される。ある一定のより強固なオペレーティング・システムは、マイクロプロセッサ・システムのコンテキストにおけるフォルトトレラント機能で構成される。この型のマイクロプロセッサ・システムは、ハードウェアまたはソフトウェアによって検出された故障から回復するためにソフトウェアをイネーブルすることによってフォルトトレラント環境を供給する。例えば、米国特許第 4,817,091号公報は、各プロセッサが、継続動作の表示をそれにより供給すべく、ソフトウェア制御の下で、システムのプロセッサのそれぞれ（それ自身を含む）に周期的にメッセージを送るようなマルチプロセッサ・システムを教示する。プロセッサのそれぞれは、その通常タスクを実行することに加えて、別のプロセッサに対してバックアップ・プロセッサとして動作する。バックアップ・プロセッサの一つが同胞（シブリング）プロセッサからのメッセージ表示を受信することに失敗した場合には、それ自身のタスクを実行することに加えて、その同胞（いま動作不能であると考えられる）の動作を取って代わる。あまり強固でないソフトウェアまたはオペレーティング・システムを用いている（即ち、検出された故障から回復するための本質的機能がない）、他のフォルトトレラント技術は、ハードウェア及び検出された誤りから回復すべく動作する論理回路で設計される。

【００４０】本発明は、両方の型のソフトウェアに対するハードウェア・プラットフォームを供給することに指向される。それゆえに、強固なオペレーティング・システムが利用可能であるときには、処理システム１０は、ＣＰＵ１２Ａ及び１２Ｂのそれぞれが独立したファッションで動作するような“シンプレックス”モードで動作すべく構成することができる。ＣＰＵｓ１２は、ＣＰＵ内部データ経路の種々の重要な地点に誤り検査回路素子を有して構成される。ルータ１４は、システム１０で相互接続されうる種々のＣＰＵｓ１２間でプロセッサ間通信を供給すると共に、システムのＣＰＵからＩ／Ｏパケット・インターフェイス１６によって制御される装置への通信ルートを供給する。誤りが検出されたときには、その誤りから回復する役割は、ほとんどの場合、ソフトウェアに残される。代替的に、あまり強固でないオペレーティング・システム及びソフトウェアに対して、処理システム１０は、一対のＣＰＵｓ（例えば、ＣＰＵｓ１２Ａ，１２Ｂ）が同期された、ロックーステップ・ファッションで動作すべく図１に示すように一緒に結合されるように“デュプレックス”モードで動作すべく構成されているハードウェア・ベース・フォルトトレランスを供給し、実質的に同じ瞬間に同じ命令を実行する。それゆえに、各ＣＰＵは、他のチェックとして動作する。ＣＰＵｓ１２の一つが故障を生じる場合には、それは、“フェイルーファースト”でかつ誤りがシステムの残りに拡散しかつそれらを汚染することが許容される前にシャット・ダウンする。他のＣＰＵ１２は、二つのタスクを実行すべく動作を継続する。次に、デュプレックス・モード動作は、システム・ハードウェアに故障の効果をマスクさせる。

【００４１】データ及び指令記号は、９ビット・データ及び指令記号を含んでいるメッセージ・パケットによって種々のＣＰＵｓ１２とＩ／Ｏパケット・インターフェイス１６との間で伝達される。ＣＰＵ１２の設計を簡略化するために、プロセッサ２０は、外側のエンティティ（例えば、別のＣＰＵ１２またはＩ／Ｏパケット・インターフェイス１６を介するＩ／Ｏ装置）と直接通信することから排除される。それよりも、見て分かるように、プロセッサは、メモリにデータ構造を構成しかつインターフェイス装置２４に制御を転換する。各インターフェイス装置２４は、メモリからのデータ構造をアクセスし

かつメッセージ・パケットに含まれた情報による宛先への通信に対して適切なＸまたはＹポートを介してそれらを伝送するための直接メモリ・アクセス（ＤＭＡ）機能のフォームを供給すべく構成されたブロック転送エンジン（ＢＴＥ；図９）を含む。処理システム１０の設計は、ＣＰＵのメモリ２８に外側のソース（例えば、ＣＰＵ１２ＢまたはＩ／Ｏ装置）によって読み取られるかまたは書き込まれることを許容する。この理由で、ＣＰＵ１２のメモリ２８の外部使用が許可されることを確実にすべく注意が払われなければならない。それゆえに、メモリ２８へのアクセスは、アクセス要求が来た場所、要求されたアクセスの型、要求されたアクセスの位置、等のようなファクターを試験することによりアクセスを許容するかまたは排除するアクセス妥当性検査（確認）機構によって保護される。アクセス妥当性検査は、以下の図１６～図２０の説明中に記載されるアクセス妥当性検査表（ＡＶＴ）論理回路によって実施される。

【００４２】本発明の種々の態様が、ルータ１４を介してＩ／Ｏパケット・インターフェイス１６とＣＰＵｓ１２との間で伝送されるデータ及び指令パケットの構成を用いる。従って、処理システム１０の構成の説明を続ける前に、ＴＮｅｔリンクＬ上で送信されかつルータ１４によって送られる（ルート付けられる）データ及び指令記号及びパケットの構成をまず理解することは、有利である。

【００４３】－　パケット構成：４つの基本メッセージ・パケット型がＣＰＵｓ１２とシステムの周辺装置１７との間に指令記号及びデータを伝達するために用いられる。図５（ａ）～図５（ｄ）は、そのパケットのフィールドの分解と共に、一つのメッセージ・パケットの構成を示す；図６～図８は、他の３つのパケット型の構成を示す。ＴＮｅｔエリア・ネットワーク上に書込みデータを伝達するために用いられるメッセージ・パケット型は、ＨＡＤＣパケットとして識別され、かつ図５（ａ）に示されている。図示するように、ＨＡＤＣパケットは、４つのフィールドを有する：８バイト・ヘッダ・フィールド、４バイト・データ・アドレス・フィールド、Ｎバイト・データ・フィールド（より大きなデータは、単一パケットによって移動することができるということが明らかであるが、Ｎは、６４の最大値であるのが好ましい）、及び４バイト・サイクリック冗長チェック（ＣＲＣ）フィールド。図５（ｂ）に詳細に示される、ヘッダ・フィールドは、３バイト宛先ＩＤを含み、メッセージ・パケットの最終宛先を識別する；　メッセージ・パケットのソースまたは送信者、トランザクションの型（例えば、読取りまたは書込み動作）及びメッセージ・パケットの型（例えば、データに対する要求か否か、またはデータ要求への応答か否か）を識別する３バイト・ソースＩＤ。宛先ＩＤは、４つのサブフィールドを含む：　宛先が配置される“領域”を特定するための領域ＩＤを含む１４ビット・サブフィールド；識別された領域内の宛先を特定している、装置ＩＤを含んでいる６ビット・サブフィールド；及び二つの経路間を選択するために用いられる経路選択（Ｐ）ビット；及び更なる拡張のために確保された３ビット。同様に、ソースＩＤは、３つのサブフィールドを有する；送信者の領域を識別する、１４ビット領域ＩＤ；その領域内の送信装置を識別する、６ビット装置ＩＤ；及び　トランザクションの型を識別する、上記したような、４ビット型サブフィールド。更に、制御フィールドは、９ビット指令／データ“記号”によるメッセージ・パケットの添付したデータ・フィールドに含まれるデータの量を特定する。（各記号は、以下に示すように、指令記号としてデータバイトを、またはその逆を、表わすことができる、単一ビット誤りに対して保護すべく９ビット量としてコード化されたデータの８ビット・バイト誤りである。）

【００４４】図５（ｂ）に特に示さないのは、宛先フィールドが３つのサブフィールド、即ち、領域(Region)（１４ビット）、装置(Device)（６ビット）及び経路選択(Path Select)（１ビット）に分割されるという事実である。領域フィールドの情報は、メッセージ・パケットの宛先を含んでいる特定の“領域”を識別する。領域の概念は、ルータ１４のアーキテクチャの説明に関連して以下に説明する。６ビット装置フィールドは、メッセージ・パケットの最終宛先である特定の装置（例えば、装置１７、ＣＰＵ１２、またはＭＰ１８）を識別する。領域及び装置フィールドは、宛先を漸増的かつ一意に識別する。経路選択ビットとして確保されたビットは、メッセージ・パケットの宛先を含んでいる（図１に示したような）二つの“側”ＸまたはＹの一つまたは他のものを識別すべく動作する。経路選択ビットは、メモリ・アクセス妥当性検査（図１６及び図１７）及びルータのポート選択動作（図３１）に関連して以下に更に説明する。残りの３ビットは、必要により更なる拡張のために確保されている。４バイト・データ・アドレス・フィールドは、図５（ｃ）に詳細に示される。アドレス・フィールドは、ＨＡＤＣパケットの場合に、データの添付したＮバイトが書込まれる宛先の仮想位置を識別する。例えば、メッセージ・パケットのソースが、ＣＰＵ１２のメモリ２８に書込まれるべきデータを含んでいるＩ／Ｏ装置１７であるならば、データ・アドレス・フィールドは、データが書込まれるべきメモリ２８における位置を識別しているアドレスを含む。（見て分かるように、ＣＰＵｓに対してデータ・アドレスは、ＡＶＴ論理回路（図１６）によってメモリ２８をアクセスするために実際に用いられる物理アドレスに変換される。Ｉ／Ｏパケット・インターフェイス１６は、同様な妥当性検査及び変換機構を有する。）アドレス・フィールドがＣＰＵ１２のメモリ位置を識別するときには、フィールドは、二つのサブフィールドを含む：アドレス・フィールドの上

位２０ビットは、２０ビット・メモリ頁数を形成する；残りの１２ビットは、メモリ頁にオフセットを形成する。頁数は、妥当性検査情報を含むエントリを含んでいる表の中へのインデックス（索引）としてＡＶＴ論理回路（図１６）によって用いられる。

【００４５】説明したように、ＨＡＤＣメッセージ・パケットは、処理システム１０の端末装置（例えばＣＰＵ１２）間で書込みデータを伝達すべく動作する。しかしながら、他のメッセージ・パケットは、それらの機能及び使用により、異なって構成されうる。それゆえに、図６は、ヘッダ、アドレス、及びＣＲＣフィールドだけを含んでいるＨＡＣメッセージ・パケットを示す。ＨＡＣパケットは、システム・コンポーネント（例えば、Ｉ／Ｏ装置１７）に読取りデータ要求を送信するために用いられる。図７は、８バイト・ヘッダ・フィールド、Ｎバイト・データ・フィールド（再度、いかなる整数でもありうるが、Ｎは、６４までである）、及び４バイト・ＣＲＣフィールドを有している、ＨＤＣ型のメッセージ・パケットを示す。ＨＤＣメッセージ・パケットは、要求されたデータのリターン（戻り）を含む、読取り要求への応答を伝達するためである。図８は、８バイト・ヘッダ、及び４バイトＣＲＣだけを含んでいる、ＨＣメッセージ・パケットを示す。ＨＣメッセージ・パケットは、データを書込むための要求に肯定応答するために用いられる。

【００４６】－　インターフェイス装置：Ｘ及びＹインターフェイス装置２４（即ち、２４ａ及び２４ｂ－図４）は、ＣＰＵ１２内の３つの主な機能を実行すべく動作する：プロセッサ２０をメモリ２８にインターフェイスすること；プロセッサに対して透過的に動作するが、その制御下で、Ｉ／Ｏサービスを供給すること；及び外側ソースからのメモリ２８へのアクセスに対する要求の妥当性を検査すること。まずインターフェイス機能に関して、Ｘ及びＹインターフェイス装置２４ａ，２４ｂは、読取られた／書込まれたデータのフェイルーファースト検査を含むような方法でデータを書込みかつ読取るためにプロセッサ２０ａ，２０ｂをメモリ・コントローラ（Ｍｃｓ２６ａ，２６ｂ）及びメモリ２８にそれぞれ伝達すべく動作する。例えば、書込み動作は、後で検索（読取り）されたときに、適当なデータが検索されるだけでなく、適切なアドレスから検索されたことが分かるように、メモリ２８に書込まれたデータだけでなく、そのデータが書込まれる位置のメモリ・アドレスも、見て分かるように、カバーする誤り訂正コード（ＥＣＣ）を発生すべくそのインテグリティ（及び同時に、インターフェイス装置２４が動作すること）を確実にすべく書込まれるべきデータをクロス・チェックするために協同している二つのインターフェイス装置２４ａ，２４ｂを有する。

【００４７】Ｉ／Ｏアクセスに関して、プロセッサ２０は、入力／出力システムと直接伝達するための機能を備えていない；それよりも、それらは、メモリ２８にデータ構造を書込み、次にそれらのデータ構造を検索すべく直接メモリ・アクセス（ＤＭＡ）動作を実行するインターフェイス装置２４へ制御を渡し、そして所望の宛先への伝達のためにＴＮｅｔにそれらを渡さなければならない。（宛先のアドレスは、データ構造それ自体に示される。）

Ｘ及びＹインターフェイス装置２４の第３の機能である、メモリ２８へのアクセス妥当性検査は、インターフェイス装置によって保守されるアドレス妥当性検査及び変換（ＡＶＴ）表を用いる。ＡＶＴ表は、アクセスが許可された各システム・コンポーネント（例えば、Ｉ／Ｏ装置１７、またはＣＰＵ１２）に対するアドレス、許可されたアドレスの型、及びアクセスが許可されるメモリの物理位置を含む。表は、入力メッセージ・パケットに含まれたアドレスが仮想アドレスなので、アドレス変換を実行するための手段にもなる。これらの仮想アドレスは、メモリ２８をアクセスするためにメモリ制御装置２６により認識可能な物理アドレスにインターフェイス装置によって変換される。

【００４８】図９を参照すると、ＣＰＵ１２ＡのＸインターフェイス装置２４ａの簡略化されたブロック図が示されている。コンパニオンＹインターフェイス装置２４ｂ（並びにＣＰＵ１２Ｂ、または他のＣＰＵ１２のインターフェイス装置２４）は、実質的に同じように構成される。従って、インターフェイス装置２４ａの説明が処理システム１０の他のインターフェイス装置２４にも同様に適用されるということが理解されるであろう。図９に示すように、Ｘインターフェイス装置２４ａは、プロセッサ・インターフェイス６０、メモリ・インターフェイス７０、割込み論理回路８６、ブロック転送エンジン（ＢＴＥ）８８、アクセス妥当性検査及び変換論理回路９０、パケット送信機９４、及びパケット受信機９６を含む。

【００４９】－　プロセッサ・インターフェイス：プロセッサ・インターフェイス６０は、プロセッサ２０ａとＸインターフェイス装置２４ａとの間の情報フロー（データ及び指令）を処理する。６４ビット・アドレス及びデータ・バス（ＳｙｓＡＤ）２３ａと９ビット指令バス２３ｂとを含んでいる、プロセッサ・バス２３は、プロセッサ２０ａ及びプロセッサ・インターフェイス６０を互いに結合する。ＳｙｓＡＤバス２３ａは、通常の時分割ファッションで、メモリ・アドレス及びデータを運ぶと同時に、指令バス２３ｂは、指令及びデータ識別子情報（ＳｙｓＣｍｄ）を運び、ＳｙｓＡＤバス２３ａ上で実質的に同時に運ばれた指令を識別しかつ認定する。プロセッサ・インターフェイス６０は、プロセッサ・インターフェイスのメモリまたは制御レジスタに読取り／書込みを渡すためにプロセッサ装置２０ａによって発行さ

れた指令を解釈すべく動作する。更に、プロセッサ・インターフェイス６０は、（メモリ・コントローラ２６を介して）メモリ２８へのアクセスに対してアドレス及びデータをバッファする一時記憶装置（図示省略）を含む。メモリから読取ったデータ及び指令情報は、プロセッサ装置２０ａへの途中で同様にバッファされ、かつプロセッサ装置がそれを受容する準備ができたときに利用可能になる。更に、プロセッサ・インターフェイス６０は、Ｘインターフェイス装置２４ａに対して必要割込み信号を生成すべく動作する。

【００５０】プロセッサ・インターフェイス６０は、双方向性６４ビット・プロセッサ・アドレス／データ・バス７６によってメモリ・インターフェイス７０及び構成レジスタ７４に接続される。構成レジスタ７４は、Ｘインターフェイス装置２４ａの他のコンポーネントに含まれる種々の制御レジスタの記号的表現であり、かつそれらの特定コンポーネントを説明するときに説明する。しかしながら、図９に特に示されていないが、構成レジスタ７４の種々のものがＸインターフェイス２４ａを実施するために用いられる論理回路の他のもの中に分散するという事実により、プロセッサ・アドレス／データ・バス７６は、同様にそれらのレジスタに対する読取りまたは書込みに結合される。構成レジスタ７４は、プロセッサ２０ａに対して読取り／書込みアクセス可能である；それらは、Ｘインターフェイス装置を“個人専用化”させる。例えば、一つのレジスタは、ＣＰＵ１２Ａから生ずるメッセージ・パケットのソース・アドレスを形成するために用いられる、ＣＰＵ１２Ａのノード・アドレスを識別する；別の、読取り可能専用は、インターフェイス装置２４の固定識別番号を含み、かつ他のレジスタは、例えば、（データ構造及びＢＴＥ指令／制御語が配置される）ＢＴＥ８８、（メッセージ・パケットを介して受信した外部で生成された割込みについての情報を含む割込みキューを指している）割込み論理回路８６、またはＡＶＴ論理回路９０によって用いることができるメモリの領域を画定する。他のレジスタは、割込み論理回路８６による割込みポスティング（記入）に対して用いられる。レジスタの多くは、それらを採り入れている論理回路コンポーネント（例えば、割込み論理回路８６、ＡＶＴ論理回路９０、等）を説明するときに以下に更に説明する。

【００５１】メモリ・インターフェイス７０は、二つの３６双方向ビット・バス２５ａ，２５ｂを含むバス２５によってＸインターフェイス装置２４ａをメモリ・コントローラ２６（及びＹインターフェイス装置２４ｂ；図４参照）に結合する。メモリ・インターフェイスは、プロセッサ装置２０、ＢＴＥ８８、及びＡＶＴ論理回路９０からのメモリ・アクセスに対する要求間を調停(arbitrate)すべく動作する。プロセッサ装置２０ａからのメモリ・アクセスに加えて、メモリ２８は、例えば、Ｉ／Ｏ装置１７からプロセッサ装置２０ａによって読取られるべく要求されたデータを記憶するために、処理システム１０のコンポーネントによってもアクセスされうるか、またはメモリ２８は、プロセッサ装置によってメモリに予めセットアップされたＩ／Ｏデータ構造に対してもアクセスされうる。これらのアクセスは、全て非同期なので、それらは、調停されなければならず、かつメモリ・インターフェイス７０は、この調停を実行する。メモリ２８からアクセスされたデータ及び指令情報は、メモリ読取りバス８２によってメモリ・インターフェイスからプロセッサ・インターフェイス６０に、並びに割込み論理回路８６、ブロック転送エンジン（ＢＴＥ）８８、及びアクセス妥当性検査及び変換（ＡＶＴ）論理回路９０に結合される。以下に詳述するように、データは、倍長語量でメモリ２８に書込まれる。しかしながら、Ｘ及びＹインターフェイス装置２４ａ及び２４ｂの両方のメモリ・インターフェイス７０は、（６４ビット）倍長語を形成しかつバス２５に適用し、各メモリ・インターフェイス７０は、その６４ビット倍長語量の３２ビットだけを書込む役割を果たす；メモリ・インターフェイス７０によって書込まれない３２ビットは、それらが誤りに対して同じ３２ビットと比較されるコンパニオン・インターフェイス装置２４によってメモリ・インターフェイスに結合される。

【００５２】ちょっと脇道にそれると、図１～図３の処理システムでは、割込みは、特定割込み型を伝達するために専用信号回線の従来技術を用いるよりも、メッセージ・パケットとして送信される。割込み情報を含んでいるメッセージ・パケットが受信されるときには、その情報は、ＣＰＵ１２Ａの内部で生成された割込みと共に、プロセッサ２０による作用に対して処理しかつポスティングするために割込み論理回路８６に運ばれる。内部で生成された割込みは、割込みの原因を示している、（割込み論理回路８６の内部の）レジスタ７１にビットをセットする。次に、プロセッサ２０は、割込みにより読取りかつ作用することができる。割込み論理回路は、以下により完全に説明する。Ｘインターフェイス装置２４ａのＢＴＥ８８は、直接メモリ・アクセスを実行すべく動作し、かつプロセッサ２０に外部資源をアクセスさせる機構を供給する。ＢＴＥ８８は、プロセッサ２０に透過である、Ｉ／Ｏ要求を生成すべくプロセッサ２０によってセットアップし、かつ要求が終了したときにプロセッサに知らせることができる。ＢＴＥ論理回路８８は、以下に更に説明する。入力メッセージ・パケットに含まれるメモリ・アクセスに対する要求は、ＡＶＴ論理回路９０によって妥当性が検査される。アクセス要求の妥当性検査は、要求のソースの識別、要求されたアクセスの型を含んでいる、種々の許可により行われる。更に、ＡＶＴ論理回路は、要求が適切に確認されたときに実際のアクセスにするために用いることができる物理メモリ・ア

ドレスに、メッセージ・パケットに含まれたときには仮想アドレスである、アクセスが所望である（受信したメッセージ・パケットに含まれる）メモリ・アドレスを変換する。また、ＡＶＴ論理回路９０は、以下により詳細に説明する。

【００５３】ＢＴＥ論理回路８８は、送られるべきデータ及び／又は指令記号をパケット送信機９４に供給すべくＡＶＴ論理回路９０に関連して動作する。次に、パケット送信機９４は、メッセージ・パケット・フォームにＢＴＥ及びＡＶＴ論理回路８８，９０から受信した情報をアセンブルし、それらが送信できるまでそれらをバッファする。更にまた、ＢＴＥ及びＡＶＴ論理回路８８，９０は、入力メッセージ・パケットを受信し、解釈しかつ処理すべくパケット受信機９６で動作し、必要によりそれらをバッファし、かつメモリ２８に記憶するために必要な８バイト幅フォーマットにそれらを変換する。プロセッサから発生されたトランザクション要求を含んでいる出力メッセージ・パケット（例えば、Ｉ／Ｏ装置からブロック・データを要求している読取り要求）は、要求トランザクション論理回路（ＲＴＬ）１００によって監視される。ＲＴＬ１００は、要求が所定の期間内に応答するかどうかを確認するアウトバウンド（出発）要求に対するタイムーアウト・カウンタを供給する；もしそうでなければ、ＲＴＬは、要求が受け入れられなかったことをプロセッサ２０に報告すべく（割込み論理回路８６によって処理されかつ報告される）割込みを生成する。加えて、ＲＴＬ１００は、応答を確認する。ＲＴＬ１００は、応答に対するアドレスを保持し、かつそれが応答を位置決めできるようにプロセッサ２０に既知な位置で応答をメモリ２８に配置することができるように応答が受信されたときにこのアドレスをＢＴＥ８８に進める。

【００５４】ＣＰＵｓ１２のそれぞれは、説明するように、多数の方法で検査される。一つのそのような検査は、各ＣＰＵのインターフェイス装置２４ａ，２４ｂの動作の前進(on-going)監視である。インターフェイス装置２４ａ，２４ｂは、ロックーステップで動作するので、同期性検査は、それらの内部状態のあるものの連続比較によりペアになったインターフェイス装置２４ａ，２４ｂの動作状態を監視することによって実行できる。このアプローチは、ＣＰＵ１２Ａの装置２４ａに含まれる状態マシン（図示省略）の１ステージを使用し、かつそのステージによって想定された各状態をインターフェイス装置２４ｂのその同じ状態マシン・ステージと比較することによって実施される。インターフェイス装置２４の全ての装置は、それらの動作を制御すべく状態マシンを用いる。従って、インターフェイス装置２４とＭＣ２６との間のデータ転送を制御するメモリ・インターフェイス７０の状態マシンを用いるのが好ましい。それゆえに、インターフェイス装置２４ａのメモリ・インターフェイス７０に用いられる状態マシンの選択されたステージが選択される。また、一つのインターフェイス装置２４ｂの状態マシンの同じステージも選択される。二つの選択されたステージは、インターフェイス装置２４ａ，２４ｂ間で伝達されかつインターフェイス２４ａ，２４ｂの両方に含まれる比較回路によって受信される。インターフェイス装置が互いにロックーステップで動作すると、状態マシンは、同じ同一状態を通って同様にマーチし、実質的に同じ瞬間で各状態を想定する。インターフェイス装置が誤りに遭遇するか、または故障したならば、そのアクティビティは、インターフェイス装置が発散する原因となり、かつ状態マシンは、異なる状態を想定する。また、状態マシンから比較回路に伝達された選択ステージが異なる場合もある。この相違は、比較回路が、そのＣＰＵのインターフェイス装置２４ａ，２４ｂがもはやロックーステップでないＣＰＵ１２Ａ（または１２Ｂ）の注意を喚起する“損失ｓｙｎｃ(lost sync)”誤り信号を発行し、かつそれに応じて作用する原因となる。この技術の一例は、Humphrey, et. al. により発明されかつ本願発明の出願人にアサインされた米国特許第 4,672,609号公報に見ることができる。

【００５５】図９に戻ると、ＣＰＵ１２ＡのＸインターフェイスのパケット受信機９６は、Ｘポートだけをサービスすべく機能し、サブプロセッサシステム１０Ａ（図１）のルータ１４Ａによって送信されたそれらのメッセージ・パケットだけを受信する。Ｙポートは、コンパニオン・サブプロセッサシステム１０Ｂのルータ１４Ｂからメッセージ・パケットを受信すべくＹインターフェイス装置２４ｂによってサービスされる。しかしながら、両方のインターフェイス（並びにＭｃｓ２６及びプロセッサ２０）は、示されたように、基本的に構造及び機能において両方が実質的に同一である互いのミラー・イメージである。この理由で、一つのインターフェイス装置（例えば、２４ａ）によって受信された、メッセージ・パケット情報は、コンパニオン・インターフェイス装置（例えば、２４ｂ）へも処理するために渡されなければならない。更に、インターフェイス装置２４ａ，２４ｂの両方は、ＸまたはＹポートからの送信に対して同じメッセージ・パケットをアセンブルするので、関連ポート（例えば、Ｙポート）から実際に伝達されるインターフェイス装置（例えば、２４ｂ）によって送信されるメッセージ・パケットは、誤りに対するクロスーチェックのために他のインターフェイス装置（例えば、２４ａ）にも結合される。これらの特徴は、図１０及び１３に示されている。

【００５６】－ パケット受信機：図１０を参照すると、Ｘ及びＹインターフェイス装置２４ａ，２４ｂのパケット受信機９６（９６ｘ，９６ｙ）の受信部分は、拡大して示されている。示したように、各パケット受信機９６ｘ，９６ｙは、ＴＮｅｔリンク3.2の対応している

ものを受信すべく結合されたクロックｓｙｎｃ（ＣＳ）ＦＩＦＯ１０２を有する。ＣＳ　ＦＩＦＯｓ１０２は、パケット受信機９６の局所クロックに入力指令／データ記号を同期すべく動作し、それらをバッファして、それをマルチプレクサ（ＭＵＸ）１０４に渡す。しかしながら、Ｘポート及びＸインターフェイス２４ａのパケット受信機９６ｘで受信した情報は、ＭＵＸ１０４ｘに渡されることに加えて、クロスーリンク接続36_xによりＹインターフェイス装置２４ｂのパケット受信機９６ｙのＭＵＸ１０４ｙに結合されるということに注目する。同様なファッションで、Ｙポートで受信した情報は、クロスーリンク接続36_yによりＸインターフェイス装置２４ａに結合される。この方法で、対応Ｙ，Ｙインターフェイス装置によりＸ，Ｙポートの一つで受信された情報パケットの指令／データ記号は、両方が同じ情報を処理しかつインターフェイス装置２４の他のコンポーネント及び／又はメモリ２８に伝達するように他のものに渡される。

【００５７】図１０を続けると、どのポートＸ，Ｙがメッセージ・パケットを受信しているいかにより、ＭＵＸｓ１０４は、インターフェイス装置２４の記憶及び処理論理回路１１０への通信に対してＣＳ　ＦＩＦＯｓ１０２ｘ，１０２ｙの一つまたは他のものの出力のいずれかを選択する。各９ビット記号に含まれる情報は、その符号化が図１４を参照して以下に説明される、指令またはデータ情報の８ビット・バイトである。記憶及び処理論理回路１１０は、まず９ビット記号を８ビットデータまたは指令バイトに変換し、６４ビット倍長語としてバイトを編成し、よのように形成された倍長語を入力パケット・バッファ（特に示していない）に渡す。入力パケット・バッファは、それをメモリ・インターフェイス７０、並びにＡＶＴ論理回路９０及び／又はＢＴＥ８８に渡すことができるまで受信した情報を一時的に保持する。パケット受信機９６は、それぞれメッセージ・パケットのＣＲＣを検査するＣＲＣチェッカ論理回路１０６を含む。各ＣＲＣチェッカ論理回路１０６は、どのポート（ＸまたはＹ）がメッセージ・パケットを受信するかに関係なく、両方の受信機９６ｘ，９６ｙが受信したメッセージ・パケットのＣＲＣを検査するように配置されるということに、特に、注目する。この特徴は、フォルト分離機能を有する。この受信ステージで検査されるけれども、他のものからではなく一つの受信機からのＣＲＣ誤り表示は、二つの受信機間のインターフェイスまたは誤りを発行している受信機の論理回路における問題を示す。それゆえに、フォルトは、受信しているＣＳ　ＦＩＦＯの出力からの経路のその部分に対して少なくとも最初のうちは分離することができる。

【００５８】図示されていないのは、ＣＳ　ＦＩＦＯｓ１０２ｘ，１０２ｙの出力は、ＭＵＸ１０４に加えて指令復号装置にも結合されるという事実である。指令復号装置は、指令記号を認識すべく（以下に説明する方法でデータ記号からそれらを区分すべく）動作し、受信機制御装置、パケット受信機動作を制御すべく機能する状態マシン・ベース素子に印加される指令信号をそれから生成すべく指令記号を復号する。上述したように、パケットは、周期冗長検査（ＣＲＣ）値によって誤り保護される。それゆえに、受信したパケットのＣＲＣ情報がＭＵＸ１０４の出力に現れるときには、記憶制御装置の受信機制御部分は、ＣＲＣ記号を計算するためにＣＲＣ検査論理回路１０６をイネーブルし、同時に、データ記号は、受信したＣＲＣに対する生成された量をメッセージ・パケットと続いて比較すべく受信される。可能な誤りがパケット受信機９６への送信中に生じたことを示している、ミスマッチが存在するならば、ＣＲＣ検査論理回路１０６は、割込みレジスタ（割込みレジスタ２８０；図２１）をセットするために用いられる誤り割込み信号（ＢＡＤＣＲＣ）を発行しかつパケットは、廃棄される。しかしながら、パケット・ヘッダは、後の試験のために割込みキューにセーブされる。

【００５９】以下に更に説明するように、ＣＳ　ＦＩＦＯｓは、インターフェイス装置２４のパケット受信機９６においてだけでなく、ルータ１４及びＩ／Ｏパケット・インターフェイス１６の各受信ポートでも見出される。しかしながら、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，１４Ｂ（即ち、ポート１及び２）を接続するＴＮｅｔリンクＬから記号を受信するために用いられたＣＳ　ＦＩＦＯｓは、ルータ１４の他のポート上で用いられるもの、及びＣＰＵ１２に直接接続されていない他のルータ１４とは多少異なる。換言すると、周波数ロックド(frequency locked)クロッキングを用いている素子間で記号を伝達するために用いられたＣＳ　ＦＩＦＯｓは、近周波数クロッキングを用いている素子間で記号を伝達するために用いられたものとは異なる。また、以下の説明は、ＣＳ　ＦＩＦＯｓは、近周波数モード（即ち、送信及び受信素子のクロック信号は、同じである必要はないが、所定のトレランス内であることが期待される）で動作している素子間でＴＮｅｔリンクＬ上の情報を転送することにおける重要部分の役をするということも明らかにする。しかし、ＣＳＦＩＦＯｓは、一対のサブプロセッサシステムがデュプレックス・モードで動作しかつサブプロセッサシステム１０Ａ，１０Ｂの二つのＣＰＵｓ１２Ａ及び１２Ｂが同期された、ロックーステップで動作し、同時に同じ命令を実行するときには、より重要な部分の役をし、かつ固有の機能を実行する。この後モードで動作しているときには、ルータ１４Ａまたは１４Ｂの一つからＣＰＵｓ１２Ａ及び１２Ｂに送信された情報が、同期、ロックーステップ動作を維持するために本質的に同時に両方のＣＰＵｓによって受信されることは、絶対必要である。ルータ１４Ａ及び１４Ｂのクロッキング・レジムがＣＰＵｓ１２Ａ及び１２Ｂのものに

正確に同期されることを確実にするのは、周波数封じ込みクロッキングを用いているときでさえも、非常に難しいので、これは、不幸にも、容易なタスクではない。ＣＰＵｓ１２のパケット受信機９６において、ＣＰＵ１２に記号を送信するために用いられるルータ１４のクロックとそれらの記号を受信すべくインターフェイス装置２４によって用いられるクロックとの間の可能な相違を収容する(accommodate) ことは、ＣＳ　ＦＩＦＯｓ１０２の機能である。

【００６０】ＣＳ　ＦＩＦＯ　１０２の構造は、図に、説明の目的で、図式的しめされている；ＣＳ　ＦＩＦＯの好ましい構造は、図１２に示されている。再び、ＣＳＦＩＦＯに対してここで参照がなされるときには、特に示さない限り、図１１を参照して説明される機能及び動作、及び図１２に示された構造を有している構造を参照することを意図するということが理解されるべきである。従って、図１１のＣＳ　ＦＩＦＯの説明は、その性質において一般的であることを意図しており、そのように理解されるべきである。更に、上記したように、周波数ロックド動作に用いられるＣＳ　ＦＩＦＯｓのあるものは、近周波数動作で用いられるものと異なるが、以下の説明は、両方に適用する。その説明に続いては、近周波数環境における動作のためにＣＳ　ＦＩＦＯの一般的な構成になされなければならない変更の説明である。図１１に示すのは、パケット受信機９６ｘのＣＳ　ＦＩＦＯ　１０２ｘである。ＣＳ　ＦＩＦＯ　１０２ｙは、ＣＳ　ＦＩＦＯ　１０２ｘの以下の説明がＣＳＦＩＦＯ　１０２ｙに同様に適用されることが理解されるような実質的に同一の構成及び動作である。図１１では、ＣＳ　ＦＩＦＯ　１０２ｘは、ルータ１４Ａ（図1）の送信（Ｘｍｔ）レジスタ１２０から送信される９ビット指令／データ記号及びルータからの添付送信クロック（Ｔ＿Ｃｌｋ）を受信すべくＴＮｅｔリンク32_xによって結合されて示される。図１１の破線Ｂは、対応ＴＮｅｔリンク32_xの一端における送信エンティティ（ルータ１４Ａ）とＣＰＵ１２Ａの受信エンティティ、パケット受信機９６ｘとの間のクロック境界を記号化する。従って、ＣＳ　ＦＩＦＯ　１０２ｘは、記憶キュー１２６へ渡される前にそれらが一時的に保持される（例えば、一つのＴ＿Ｃｌｋ期間に対して）受信（Ｒｃｖ）レジスタ１２４で9ビット記号を受信する。記憶キュー１２６は、表示及び説明の容易のために４つの位置を含んで示される。しかしながら、追加記憶位置が供給でき、かつ事実必要または所望でありうることは、この分野における当業者には、明らかである。

【００６１】受信した記号は、プッシュ・ポインタ・カウンタ１２８によって識別された記憶キュー１２６の位置で（Ｒｃｖレジスタ１２４から）ＣＳ　ＦＩＦＯ　１０２ｘ上に“プッシュ”される。プッシュ・ポインタ・カウンタ１２８は、Ｔ　Ｃｌｋによってクロックされる、バイナリ・カウンタ（２進計数器）の形であるのが好ましい。次に、受信した記号は、プル・ポインタ・カウンタ１３０によって識別される記憶キュー１２６の位置から逐次的に“プル(pull)”され、ＦＩＦＯ出力レジスタ１３２に渡される。局所クロック信号、“Ｒｃｖ　Ｃｌｋ”は、記憶キュー１２６及びＦＩＦＯ出力レジスタ１３０から記号をプルするために用いられ、（ＣＰＵ１２Ａに対して）内部で生成された信号によって生成される。ＦＩＦＯ出力レジスタ１３２からの記号は、ＭＵＸ１０４ｘに行く。ＴＮｅｔ送信に用いられたプロトコルにより、記号の一定ストリームは、常に全ての送信ポート（例えば、ＣＰＵ１２ａのＸ及びＹポート、ルータ１４ＡまたはＩ／Ｏインターフェイス１６のいずれかの送信ポートー図1）から送信される；それらは、ある一定の情況（例えば、リセット、初期化、同期化及び以下に説明するその他）の間を除いて　－　実際の指令／データ記号（例えば、パケット）またはＩＤＬＥ記号のいずれかでありうる。上述したように、ルータ１４Ａの送信レジスタ１２０に保持された各記号は、Ｒｃｖレジスタ１２４に結合され、かつルータ１４Ａによって供給されるクロック信号、Ｔ＿Ｃｌｋで、記憶キュー１２６に記憶される。逆に言えば、記号は、局所的に生成されたクロック、Ｒｃｖ　Ｃｌｋと同期して記憶キュー１２６からプルされる。これらは、実質的に同じ周波数であるにもかかわらず、二つの異なるクロック信号である。しかしながら、ＣＳ　ＦＩＦＯ　１０２ｘに入ってくる記号とＣＳ　ＦＩＦＯからプルされるその同じ記号との間に十分な時間（例えば、２～３のクロック）が存在する限り、準安定性問題が存在すべきではない。入力クロック信号（Ｔ＿Ｃｌｋ）及びＲｃｖ　Ｃｌｋが周波数封じ込みモードで動作されるときには、ＣＳ　ＦＩＦＯ１０２ｘは、決してオーバーフローまたはアンダーフローすべきでない。

【００６２】ＣＳ　ＦＩＦＯ　１０２ｘを初期化するのは、次の通りである。アウトセットでは、ルータ１４Ａは、送信クロック信号、Ｔ＿Ｃｌｋの各パルスに対するＩＤＬＥ記号を送信し、ＩＤＬＥ記号でＲｃｖレジスタ１２４、記憶キュー１２６、及びＦＩＦＯ出力レジスタ１３２を最後に充たし、ＣＳ　ＦＩＦＯ　１０２ｘをアイドル状態にリセットする。プッシュ・ポインタ・カウンタ１２８及びプル・ポインタ・カウンタ１３０は、ＳＹＮＣ指令記号の受信（及び検出）によりリセットされる。ＳＹＮＣ信号の受信は、プッシュ・ポインタ・カウンタ１２８を、記憶キュー１２６の特定の位置に対するポイントにセットする原因になる。同時に、プル・ポインタ・カウンタ１３０は、好ましくは２つの記憶位置によってプッシュ・ポインタ・カウンタのそれから離間した記憶キュー１２６の位置におけるポイントに同様にセットされる。それゆえに、公称(nominal) ２クロック遅延は、記憶キュー１２６に入ってくる記号と記憶キュー

を出て行くその同じ記号との間に確立され、それがクロック・アウトされかつMUX１０４ｘ（及び１０４ｙ）によって記憶及び処理装置１１０ｘ（及び１１０ｙ）に渡される前に記憶キュー１２６に入ってくる各記号を整定(settle)させる。送信及び受信クロックは、位相一独立であるので、公称２クロック遅延は、許容されたリセット・スキューが１クロック以下であるべく期待されるようにある所定量が＋または－の誤りを含む。

【００６３】図１２は、各組合せが記憶キュー１２６の記憶位置を形成している、マルチプレクサ／ラッチ組合せ１４０，１４２によって形成されているとして記憶キュー１２６を示している、ＣＳ　ＦＩＦＯ　１０２ｘの一つの実施を示す。ラッチ１４２は、Ｔ＿Ｃｌｋの各パルスでクロックされる。プッシュ・ポインタ・カウンタ１２８は、その関連ラッチ１４２に結合されるべくマルチプレクサ１４０の一つにｒｃｖレジスタ１２４の出力を選択させるべくデコータ１４４によって復号される。ラッチは、Ｔ＿ＣｌｋとＲｃｖレジスタを関連ラッチ１４２に伝達させるためにマルチプレクサ１４０の別のものをもたらすべくインクリメントされたプッシュ・ポインタ・カウンタとで装填される。ｒｃｖレジスタ１２４の出力を受信すべく選択されないそれらのラッチ１４２は、その代わりにＴ＿Ｃｌｋを有するラッチの内容を受信しかつ装填する。実質的に同じ時間に、プル・カウンタ１３０は、各Ｒｃｖ　Ｃｌｋで　－　ＦＩＦＯ出力レジスタ１３２に転送されかつＦＩＦＯ出力レジスタ１３２によって装填されるべく、マルチプレクサ１４６を介して、ラッチの一つの内容を選択する；プル・ポインタ・カウンタは、同時に、更新される（インクリメントされる）。

【００６４】ＣＳ　ＦＩＦＯ　１０２ｘは、一対のＣＰＵｓ１２がデュプレックス・モードで機能しているときにだけ、かつルータ１４Ａ，１４ＢとペアになったＣＰＵｓ１２Ａ，１２Ｂとの間の送信に対してだけ用いられる周波数封じ込みクロッキング（即ち、Ｔ＿Ｃｌｋ及びＲｃｖ　Ｃｌｋは、周波数において実質的に同じであるが、同相である必要なない）を実施すべく構成される（図１、図２及び図３）。ＣＰＵｓ１２と通信していない（デュプレックス・モードで機能している）ルータ１４（及びＩ／Ｏインターフェイス１６）の他のポートは、近周波数クロッキングで記号を送信すべく動作する。そうであっても、クロック同期化ＦＩＦＯｓは、近周波数クロッキングで送信された記号を受信すべくこれらの他のポートで用いられ、これらクロック同期化ＦＩＦＯｓの構造は、周波数封じ込み環境で用いられたもの、即ち、ＣＳ　ＦＩＦＯｓ　１０２のものと実質的に同じである。しかしながら、違いが存在する。例えば、記憶キュー１２６の記号位置は、９ビット幅である；近周波数環境では、クロック同期化ＦＩＦＯｓは、１０ビット幅であるキュー１２６の記号位置を用い、余分のビットは、その状態に依存して、関連記号が有効か否かを識別する、“有効”フラグである。この特徴は、この議論において更に説明する。

【００６５】ルータ１４は、記号を送信または受信すべくルータ１４のものと同じ公称周波数であるが、多少異なる実周波数を有する、他のクロック・ソースの保護（影響）の下で走る他のキャビネットの装置（例えば、他のルータまたはＩ／Ｏインターフェイス１６）と通信しているそれ自身をしばしば見出しうる。これは、近周波数情況であり、かつこれは、記号転送に対するクロッキングのこの形は、デュプレックス・モードであるときにＣＰＵ１２に直接接続するそれらのポートを除くルータ１４の全ポートで見られる。近周波数モードでは、クロック信号（例えば、一端で記号を送信するために用いられるクロック、及び他端で記号を受信するために用いられるクロック）は、他に対してサイクルを結果としてゲインする一つでゆっくりドリフトしうる。これが起きるときには、ＣＳ　ＦＩＦＯ　１０２の二つのポインタ（それぞれプッシュ及びプル・ポインタ・カウンタ１２８，１３０）は、どのエンティティ（送信機または受信機）がより速いクロック・ソースを有するかによって、より近い記憶キュー１２６の一つの記号位置または互いにもっと離れた一つの記号位置のいずれかをポイントする。このクロック・ドリフトを処理するために、二つのポインタは、周期的に効果的に再同期される。

【００６６】ＣＰＵｓ１２がペアにされかつデュプレックス・モードで動作しているときには、全ての４つのインターフェイス装置２４は、同じデータを送信しかつ同じクロック（Ｔ＿Ｃｌｋ及びＲｃｖ　Ｃｌｋ）でデータを受信すべく、ことのほか、ロックーステップで動作し、周波数封じ込みクロッキングが必要でありかつ用いられる。ＣＰＵｓ１２がシンプレックス・モードで動作するときには、それぞれは他とは独立であり、クロッキングは、近周波数であるだけが必要である。インターフェイス装置２４は、Ｒｃｖレジスタ１２４を初期化しかつ送信ルータ１４に同期するためにＳＹＮＣ指令記号との組合せで用いられるＳＹＮＣ　ＣＬＫ信号を受信する。記号転送に対して近周波数または周波数封じ込みクロッキング・モードのいずれかを用いるときには、ＣＳ　ＦＩＦＯ　１０２ｘは、ある既知の状態から開始するのが好ましい。入力記号は、パケット受信機９６の記憶及び処理装置１１０によって試験される。記憶及び処理装置１１０は、指令記号を探し、かつそれにより適切に作用する。ここで関連するのは、パケット受信機９６がＳＹＮＣ指令記号を受信するときにそれが記憶及び処理装置１１０によって復号されかつ検出されることである。記憶及び処理装置１１０によるＳＹＮＣ指令記号の検出は、ＲＥＳＥＴ信号のアサーションをもたらす。ＲＥＳＥＴ信号は、ＳＹＮＣ　ＣＬＫ信号の同期制御下で、（クロック同期化バッファを含んでいる）入力バッ

ファを所定の状態にリセットし、かつそれらをルータ１４に同期すべく用いられる。

【００６７】ルータ１４Ａ，１４Ｂの一つまたは両方のインターフェイス装置２４のＣＳＦＩＦＯｓ　１０２の同期化は、同期化を説明するセクションにおいて以下により完全に説明する。

【００６８】－　パケット送信機：各インターフェイス装置２４は、ＣＰＵ１２のＸまたはＹポートの一つだけから送信しかつそこで受信すべく割り当てられる。インターフェイス装置２４の一つが送信するときには、他は、送信されるデータを検査すべく動作する。シンプレックス・モードで動作するときでさえもＣＰＵ１２に自己検査フォルト検出及びフォルト包括機能を供給するので、これは、パケット送信機の重要な機能（特徴）である。この機能（特徴）は、Ｘ及びＹインターフェイス装置２４ａ，２４ｂのパケット送信機９４ｘ，９４ｙを、短縮形で、それぞれ表わす、図１３に示されている。両方のパケット送信機は、同じように構成され、一つ（パケット送信機９４ｘ）の説明は、特に示したことを除き、他（パケット送信機９４ｙ）に同様に適用される。図１３に示すように、パケット送信機９４ｘは、倍長語（６４ビット）フォーマットで一送信されるべきデータを、関連インターフェイス装置（ここでは、Ｘインターフェイス装置２４ａ）のＢＴＥ８８またはＡＶＴ９０から、受信するパケット・アセンブリ論理回路１５２を含む。パケット・アセンブリ論理回路１５２は、ＣＰＵ１２のＸまたはＹポートからの送信の準備ができるまで情報をバッファし、倍長語フォーマットからバイト・フォーマットにデータを変換すべくバイト・ステアリング動作を実行し、パケット・フォーマットにバイトをアセンブルし、かつＸまたはＹエンコーダ１５０ｘ，１５０ｙの一つにそれらを渡す。エンコーダ１５０の一つだけが、どのポート（ＸまたはＹ）が合成メッセージ・パケットを送信するかによって、バイトを受信する。

【００６９】８ビット・バイトを受信するＸまたはＹエンコーダ１５０は、図１４に示した９ビット指令／データ記号にそれを符号化すべく動作する。合成９ビット記号の３つの左側ビットの符号化が以下の表１の３つの最左列に示される。

【００７０】

【表１】

表　１

８Ｂ－９Ｂ　記号符号化

ＣＤＣ	ＣＤＢ	ＣＤＡ	機　能
0	0	0	指令
0	0	1	誤り
0	1	0	誤り
1	0	0	誤り
0	1	1	データ＜７：６＞＝００
1	0	1	データ＜７：６＞＝０１
1	1	0	データ＜７：６＞＝１０
1	1	1	データ＜７：６＞＝１１

【００７１】表１が示すように、図１４に関連して見ると、９ビットの高位３ビット（ＣＤＣ，ＣＤＢ，ＣＤＡ）は、記号の残りの、下位６ビット（ＣＤ５，ＣＤ４，ＣＤ３，ＣＤ２，ＣＤ１，及びＣＤ０）が（１）指令情報または（２）データとして解釈されるべきかを示すべく符号化される。結果として、３つの上位ビットＣＤＣ，ＣＤＢ，及びＣＤＡが全てゼロであるならば、９ビット記号は、指令記号としてそれにより識別され、かつ残りの６ビットは、指令を形成する。例えば、“０００ｃｃｃｃｃｃ”と表される指令／データ記号は、“ｃ”ビットが指令である、指令として解釈される。他方、指令／データ記号の、３つの上位ビットＣＤＣ，ＣＤＢ，及びＣＤＡが、データを表わす４つの値のいずれかを取るならば、それらは、残りの６ビットのデータと組合わされるべき２ビットのデータとして解釈され、それから１バットのデータを得る。残りの６ビットは、データバイトの下位ビットである。従って、“１１０００１１０１”と表される、指令／データ記号は、データ記号として解釈され、かつ“１０００１１０１”と表される１バイトのデータに変換される。上位３ビットが　００１，０１０，及び１００の形を取るならば、それは誤りである。

【００７２】指令記号からそのデータ記号を分離する３つの誤りコードは、指令及びデータ間の二つの最小ハミング距離(Hamming distance)を確立する。単一ビット誤りは、データを指令記号にまたはその逆に変えることができない。更に、（データ記号とは反対側の）指令記号の下位６ビットは、指令を含んでいる６ビット位置が正確に３つの“１”を常に含むような既知の“６つのうちの３つ”の符号に符号化される。全ての単一方向誤り、並びに指令記号のあらゆる奇数の誤りは、検出される。データにおける誤りは、指令記号をデータに変える誤りであるように、パケットＣＲＣｓを介して検出される。データを指令記号に変える誤りは、以下により完全に説明

されるように、ＣＲＣ及び／又はプロトコル違反（妨害）誤りによって検出される。ＸまたはＹエンコーダ１５０のどれがパケット・アセンブリ論理回路１５２から情報のバイトを受信するのかは、取るべき経路を示している経路ビット（Ｐ）を含んでいる、送信されるべき情報に含まれる宛先ＩＤに基づく。例えば、情報の宛先ＩＤは、それがＣＰＵ１２のＸポートを介して送られるということを提案するものと想定する。（パケット送信機９４ｘ，９４ｙの両方の）パケット・アセンブリ論理回路１５２は、その情報をＸエンコーダ１５０ｘに送る；同時に、それは、ＩＤＬＥ記号をＹエンコーダ１５０ｙに送る。（記号は、Ｘ及びＹポートから連続的に送られる：　それらは、送信される処理におけるメッセージ・パケットを形成する記号、ＩＤＬＥ記号、または制御機能を実行すべく用いられる他の指令記号のいずれかである。）

Ｘ及びＹエンコーダ１５０の出力は、マルチプレクサ１５４、１５６を含んでいる、マルチプレキシング構造に印加される。マルチプレクサ１５４の出力は、Ｘポートに接続する。（インターフェイス装置２４ｂは、マルチプレクサ１５４の出力をＹポートに接続する。）マルチプレクサ１５６は、クロスーリンク３４ｙを介して、Ｙポートに接続するマルチプレクサ１５４の出力も受信するチェッカ論理回路１６０に接続する。Ｘポート及びＴＮｅｔリンク30_xに接続するマルチプレクサ１５４の出力は、（インターフェイス装置２４ｂの）パケット送信機９４ｙのチェッカ論理回路１６０にクロスーリンク34_xによっても結合されるということに注目する。

【００７３】マルチプレクサの選択（Ｓ）入力は、構成レジスタ１６２のＸ／Ｙステージからの１ビット出力を受信する。構成レジスタ１６２は、インターフェイス装置２４に形成されるＯＬＡＰ（図示省略）を介してＭＰ１８にアクセス可能であり、かつ　インターフェイス装置２４を、特に、“個人専用”にする情報を伴って書込まれる。ここで、構成レジスタ１６２のＸ／Ｙステージは、Ｘエンコーダ１５０ｘ出力をＸポートに伝達すべくＸインターフェイス装置２４ａのパケット送信機９４ｘを構成する；Ｙエンコーダ１５０ｙの出力は、チェッカ１６０に同様に結合される。同様なファッションで、（Ｙインターフェイス装置２４ｂの）Ｙパケット送信機９４ｙの構成レジスタ１６２のＸ／Ｙステージは、マルチプレクサ１５４にＹエンコーダ１５０ｙの出力を選択させ；かつそれがＸポート送信と比較されるパケット送信機１６０のチェッカ１６０に結合されるべくＸエンコーダ１５０ｘの出力を選択させるような状態にセットされる。簡単に言うと、ＸまたはＹポートからのメッセージ・パケット送信の動作は、次の通りである。まず、示されたように、メッセージ・パケット送信が存在しないときには、Ｘ及びＹエンコーダの両方は、制御機能を実行すべく用いられるＩＤＬＥ記号または他の記号を送信する。両方のパケット送信機９４の構成レジスタ１６２のＸ／Ｙステージが上記したようにセットされると（即ち、マルチプレクサ１５４によって出力ポート（Ｘ）に伝達されたパケット送信機９４ｘのＸエンコーダ１５０ｘ；マルチプレクサ１５４によってポート（Ｙ）に伝達されたパケット送信機９４ｙのＹエンコーダ１５０ｙ）、（パケット送信機９４ｘの）Ｘエンコーダ１５０ｘからのＩＤＬＥ記号は、ＣＰＵ１２ＡのＸポートから送信され、かつ（パケット送信機９４ｙの）Ｙエンコーダ１５０ｙによって生成されたＩＤＬＥ記号は、Ｙポートから送信される。同時に、Ｘポート送信は、パケット送信機９４ｙのチェッカ１６０にクロスーリンク３４ｘによって結合され、かつそのパケット送信機のＸエンコーダ１５０ｘによって生成されたもので検査される。同じ方法で、Ｙポートから出力するＩＤＬＥ記号は、パケット送信機９４ｙから、それらがパケット送信機９４ｘのＹエンコーダ１５０ｙによって生成されたものに対して検査されるパケット送信機９４ｘのチェッカ１６０に結合される。

【００７４】この説明は、重要な事実を明らかにする：パケット送信機は、それらが正しい動作に対して監視されるためにメッセージ・パケットを送信する必要がない。逆に、メッセージ・パケット・トラフィックが存在しないときでさえも、二つのパケット・インターフェイス９４（及び、それにより、それらが関連するインターフェイス装置２４）の動作は、連続的に監視される。チェッカの一つがそれに印加されたものの間でミスマッチを検出したならば、ＥＲＲＯＲ信号がアサートされて、プロセッサ２０による適切なアクションに対して内部割込みがポスト（表示）されることを結果として生ずる。メッセージ・パケット・トラフィックは、同じような方法で動作する。パケット送信機９４のパケット・アセンブリ論理回路１５２が送信に対して情報を受信し、かつ宛先ＩＤがＸポートが用いられるべきであることを示すということを、ここで、想定する。パケット・アセンブリ論理回路は、各バイトを符号化された９ビット形に変換する、両方のインターフェイス装置９６のＸエンコーダ１５０ｘに、一度に１バイトずつ、その情報を進める。パケット送信機９４ｘのＸエンコーダ１５０ｘの出力は、ＸポートかつＴＮｅｔリンク３０ｘ、及びパケット送信機９４ｙのチェッカ１６０にマルチプレクサ１５４によって伝達される。また、パケット送信機９４ｙのＸエンコーダの出力は、それが、パケット送信機９４ｘからのもので検査される、チェッカ１６０にだけ、マルチプレクサ１５６によって、結合される。再び、インターフェイス装置２４ａ，２４ｂの動作、及びそれらが含むパケット送信機は、誤りに対して検査される。

【００７５】同じファッションで、Ｙポート・メッセージ・パケット送信が監視されるということをここで理解することができる。図９にちょっと戻り、出力メッセー

ジ・パケットがプロセッサ始動型トランザクション（例えば、読取り要求）であるならば、プロセッサ２０は、応答でメッセージ・パケットが戻されることを期待する。それゆえに、ＢＴＥ８８が送られるべきデータをメモリ２８からパケット送信機９４に転送するときには、それは、要求トランザクション論理回路１００の要求タイマ（図示省略）をセットし、その内で応答を受信すべきタイムアウト期間をマーキングすることを要求タイマに開始させる。出力要求に対する応答が受信されたときには、パケット受信機９６の応答整合(reply match) 回路は、メッセージ・パケットが応答であり、かつ要求タイマがリセットされることを決定する。宛先への顕著な要求の各数に対して一つの要求タイマ（図示省略）だけが存在する。ＢＴＥ８８がトランザクションの送信を始動するたびに、タイマがリセットされる。他方、割り当てられた時間内に応答が受信されなかったならば、要求タイマは、特定のトランザクション（例えば、読取り要求）に対する応答の欠如をプロセッサ２０にそれにより報告すべく割込み論理回路（図２１）にタイムアウト（時間切れ）信号を発行する。複数の顕著な要求が管理されることが望ましいならば、要求タイマの追加するもの－－各顕著な要求に対して一つ－－を用いることができる。

【００７６】ＣＰＵ１２Ａのメモリ２８への外部アクセスが供給されるけれども、それは、保護なしではない。メモリ２８へのアクセスに対する外部的に生成された要求は、認められ、かつ２～３の例として、要求のソースの識別、要求されたアクセスの型（例えば、読取りまたは書込み）、アクセスのメモリ領域、を含んでいる、ある一定の基準に従って許可されたときにのみ許される。また、アクセスされることが望ましいメモリ装置２８のメモリの領域は、仮想またはＩ／Ｏメモリ・アドレスによりメッセージ・パケットで識別される（それにより、仮想記憶方法を使用させる）。許可、及び許されるならば、アクセスの決定は、これらの仮想アドレスがメモリ２８の物理アドレスに変換されることを必要とする。そして、ＣＰＵ１２Ａの外部の装置または素子によって生成された割込みは、プロセッサ２０を中断すべくメッセージ・パケットを介して送信され、受信したときにメモリ２８にも書き込まれる。これら全ては、割込み論理回路及びＡＶＴ論理回路８６，９０によって処理される。ＡＶＴ論理回路装置９０は、メモリ２８へのアクセスが許可された各可能な外部ソースに対するＡＶＴエントリを含んでいる（メモリ２８のプロセッサ２０によって保守される）表を用いている。各ＡＶＴエントリは、指定のソース素子または装置、及びアクセスが認められるメモリの、特定の頁（頁は、標準的に４Ｋ（４０９６）バイトである）、または頁の一部分を識別する。一頁以上がＣＰＵ１２の外部の素子によってアクセスされるならば、素子によってアクセスされることが望ましい各頁に対してＡＶＴエントリが存在しなければならない。更に、各ＡＶＴエントリは、許可されたメモリ動作の型（例えば、書込み、読取り、または両方）の情報を含む。ＡＶＴ表は、必要でなく、かつ“期待された”メモリ・アクセスに対して用いられない。期待されたメモリ・アクセスは、Ｉ／Ｏ装置からの情報に対する読取り要求のような、ＣＰＵ１２（即ち、プロセッサ２０）によって始動されたものである。これら後者のメモリ・アクセスは、各プロセッサ始動型要求に割り当てられたトランザクション・シーケンス番号（ＴＳＮ）によって処理される。読取り要求が生成された頃に、プロセッサ２０は、読取り要求に応じて受信されることが期待されるデータに対してメモリの領域を割り当てる。この領域に対するアドレスは、読取り要求が送られるときに要求トランザクション論理回路１００によって保守されるレジスタ・ファイル（図示省略）に記憶され、かつアドレスに対するレジスタ・ファイルへのポインタは、ＴＳＮである。それゆえに、読取り要求への応答は、データを伴ってリターンし、かつ戻されたデータを記憶すべくメモリのバッファ領域のアドレスを得るためにそれが運ぶＴＳＮを用いる。

【００７７】アクセス妥当性検査は、以下のセクションでより完全に説明される。メモリ・アレー２８は、事実上、それぞれがメモリ２８に書込まれるかまたはメモリ２８から読取られる各６４ビットの倍長語の半分を管理するメモリ・コントローラ２６ａ，２６ｂによってそれぞれが管理される二つの半分に分割される。次に、メモリ・コントローラ２６ａ及び２６ｂは、各インターフェイス装置２４ａ，２４ｂのメモリ・インターフェイス７０にそれぞれ結合される。６４ビット倍長語は、“上位”ＭＣ２６ａによって書込まれる上位３２ビット（及び関連ＥＣＣ）及び“下位”ＭＣ２６ｂによって書込まれる下位３２ビット（及び関連ＥＣＣ）でメモリ２８に書込まれる。Ｍｃｓ２６ａ，２６ｂは、各々、インターフェイス装置２４ａ，２４ｂ（図９）のそれぞれのメモリ・インターフェイス７０（７０ａ，７０ｂ）からデータの３２ビット及び４ＥＣＣチェック・ビットをそれぞれ受信する。図１５を参照すると、各メモリ・インターフェイス７０は、メモリに書込まれるべき６４ビットのデータを、関連インターフェイス装置２４の、プロセッサ・インターフェイス６０からのバス８２またはＡＶＴ論理回路９０（図９参照）からのバス８３のいずれかから受信する。バス７６及び８３は、どれがＭＣＡＤバス２５に結合されるべきであるかを選択するマルチプレクサ（ＭＵＸ）８４に印加される。

【００７８】各メモリ・インターフェイス７０ａ，７０ｂは、同じで、かつ全ての、メモリに書込まれるべき６４ビットを受信するが、それぞれは、それら６４ビットのデータの半分（及びそれぞれが生成する８ビットのＥＣＣチエック・ビットの４つ）だけをＭｃｓ２６ａ，２

６ｂに転送する。Ｍｃｓ２６を駆動するために用いられない３２ビット（及びＥＣＣ論理回路８５によって生成された８ビットのＥＣＣチエック・ビットの４つ）は、それらの間のクロス検査のために各メモリ・インターフェイス７０から他に結合される。それゆに、例えば、（インターフェイス装置２４ａの）メモリ・インターフェイス７０ａは、６４ビットのデータの“上位”３２ビット（及び８ビットＥＣＣ検査語の４ビット）だけでＭＣ２６ａを駆動する。同時に、メモリ・インターフェイス７０ａは、そのコンパニオン・メモリ・インターフェイス７０ｂからデータの“下位”３２ビットを受信し、かつ比較論理回路８１によりそれ自身の下位３２ビットとそれを比較する。ミス比較が検出されたならばＥＲＲＯＲ信号がアサートされる。同様なファッションで、コンパニオン・メモリ・インターフェイス７０ｂは、メモリ２８に書込まれるべく６４ビット倍長語を伴って供給されるが、下位３２ビット（及び生成されたＥＣＣ検査ビットの４ビット）だけが用いられる。メモリ・インターフェイスは、メモリ・インターフェイス７０ａから上位３２ビットを受信し、かつ比較論理回路８１でそれらをそれ自身の上位３２ビットと比較して、ミス比較が結果として生じたならばＥＲＲＯＲ信号を発行する。

【００７９】追加の誤り検査は、各メモリ・インターフェイス７０のＥＣＣ検査回路８５によって読取り動作上で実行される。ＭＣ２６から戻された各６４ビット倍長語は、８つのＥＣＣ検査ビットを一緒に、両方のメモリ・インターフェイス７０によって受信される。データ及びＥＣＣ検査ビットは、各メモリ・インターフェイス７０のＥＣＣ論理回路８５に印加され、通常のファッションでデータのインテグリティを検査するためのシンドロームを生ずる。単一ビット誤りが検出されたならば、ＥＣＣ論理回路８５は、訂正（修正）を実行する；訂正不可能な誤りが検出されたならば、ＥＣＣ論理回路は、割込みレジスタ２８０（図２６）の状態をセットすることを結果として生ずる、誤り信号（図示省略）を発行し、かつ動作の凍結をもたらす。各メモリ・インターフェイスのＥＣＣ論理回路８５によって実施される特定のＥＣＣ検査は、１１２ビット・フィールドまでのＳＥＣ－ＤＥＣ－ＳｂＥＤアクロス(across)に対して８検査ビットを用いる。コードは、奇数列重みコード(odd column weight code)であり、あらゆる単一誤りが奇数のシンドローム・ビットを生成することを意味する。１１２の可能ビットのうち、６４は、データ、８は、検査ビットであり、４０ビットを未使用のまま残す。

【００８０】－　アクセス妥当性検査：先に示したように、ＣＰＵ１２Ａの外部の処理システム１０のコンポーネント（例えば、Ｉ／Ｏパケット・インターフェイス１６の装置、またはＣＰＵ１２Ｂ）は、メモリ２８を直接アクセスできるが、認定なしにではない。アクセス妥当性検査は、インターフェイス装置２４のＡＶＴ論理回路９０によって実施されるように、それらのメモリ位置に書込まれるべきでない他のデータで良好なデータが誤って(erroneously)または不注意に(inadvertently)上書きされることによりメモリ２８の内容が汚染されることを防ぐべく動作する。同様に、アクセス妥当性検査は、間違ったメモリ位置を不注意に読取り、それにより読み取られるデータを要求しているエンティティまたはシステム素子に誤ったデータを供給する、アクセスに対する保護も供給する。これら及び同様な理由で、アクセス妥当性検査方法は、メモリ・アクセスが適切に行われること、即ち、適切な装置が適切なメモリ位置に書込むか、または適切なメモリ位置から読取ること、を確実にすべく供給される。入力メモリ要求（即ち、読取りまたは書込み）が確認されたならば、メモリ位置のアドレスは、要求を運んでいるメッセージ・パケットのアドレス・フィールドによって運ばれるように、メモリ・アドレスにＡＶＴ論理回路によって変換される。

【００８１】メモリ２８へのアクセスは、６つ検査の全てを用いて、各インターフェイス装置２４（図９）のＡＶＴ論理回路９０によって確認される：（１）要求を運んでいるメッセージ・パケットのＣＲＣが誤りなしである、（２）メッセージ・パケットで識別された宛先（例えば、ＣＰＵ１２Ａ）が受信機のそれである、（３）メッセージ・パケットで識別された要求のソースが正しいソースである、（４）シークされたアクセスの型がアクセスを要求しているソースに対して許容される、（５）アクセスがシークされるメモリ２８の位置へのアクセスがソースに対して許容される、かつ（６）アクセスの転送サイズが所定のバウンド（上下限）内である。最初の検査は、上述したように、ＣＲＣ論理回路チェッカ１０６によって、パケット受信機９６で行われる。受信したメッセージ・パケットが不良ＣＲＣ（または、以下に示すように、それが“このパケットは、不良”（ＴＰＢ）指令記号でタグされる）を有することが見出されたならば、パケットは、廃棄され、かつアクセスは、否定される。メッセージ・パケット・ヘッダに含まれる宛先ＩＤは、パケットの宛先が正しい（即ち、ＣＰＵによって受信されたならば、適切なＣＰＵ１２が宛先として指定される）ことを確実にすべく受信素子に割り当てられた宛先ＩＤに対して比較される。ミスマッチは、パケットが何らかで誤指向され、かつパケットが再び廃棄され、そして、もちろん、アクセスが再び否定されたことを表わす。

【００８２】残りの検査は、少なくともそのメモリがアクセスされる素子のメモリへのアクセスのある形が認められた各システム素子に対してアクセス妥当性検査（ＡＶＴ）エントリ（図１８）を、メモリ２８において、保持することによってなされる。入力パケットのヘッダのアドレス・フィールドは、ソースＩＤ（ＳｏｕｒｃｅＩＤ）において識別されたシステム素子に対するＡＶＴエ

ントリを含んでいるメモリ位置へのポインタとして用いられる。ＡＶＴ論理回路は、どのアクセスがメッセージ・パケットの識別されたソースを許可されるかを決定するためにＡＶＴエントリの妥当性検査情報を用いる。それゆえに、受信したメッセージ・パケットのソースＩＤフィールドは、パケットのクリエータがＣＰＵ１２のメモリ２８へのアクセスを許可されるかどうかを決定するために用いられる。この検査は、特定のソースが特定の受信機のファシリティへのアクセスを認められるべきか否かを決定するためにパケット・ヘッダのソースＩＤフィールドをＡＶＴエントリ（ソースＩＤ）の部分と比較することを含む。シークされるアクセスの型（例えば、メモリの読取りまたは書込み）を識別している、パケットの型フィードは、シークされるアクセスの型がメッセージ・パケットによって識別されたソースを許可されるかどうか、またはパケットが求められていない応答（誤りとして削除される）であるかどうかを決定すべく検査される。

【００８３】そして、シークされたメモリ位置、及びあらゆる転送のサイズは、それらが特定のメッセージ・パケット・ソースも許可されるかどうかを見るために検査される。インターフェイス装置２４ａのアクセス妥当性検査機構、ＡＶＴ論理回路８８は、図１６に詳細に示される。ＣＰＵ１２のメモリ空間へのアクセスをシークしている入力メッセージ・パケットは、パケット受信機９６（図９）からＡＶＴ論理回路９０のＡＶＴ入力レジスタ１７０へ転送されたそれらのヘッダの選択された部分を有する。従って、ＡＶＴ入力レジスタ１７０は、ソースＩＤと、メモリ２８に書込まれるかまたはそれから読取られるべきデータの量を識別している、レングス（Ｌｅｎ）フィールドと、ＡＶＴ表エントリを含んでいるメモリ２８のエントリを指し示している、アドレス（ＡＶＴ頁#）と、ＡＶＴエントリが指し示すそのメモリ頁の中へのオフセットと、シークされたアクセスの型（Ｔｙｐｅ）とを入力メッセージ・パケットから受信する。これらの値は、ＡＶＴ入力レジスタ１７０のレジスタ・セグメント１７０ａ，１７０ｂ，．．．，１７０ｅにそれぞれ含まれる。

【００８４】ＡＶＴ入力レジスタ１７０に含まれるＡＶＴ頁数フィールドは、妥当性検査に対して必要なＡＶＴエントリのアドレスを生成すべく組合せ論理回路１７６によりそれがＡＶＴベース・レジスタ１７４の内容と組合わされるＡＶＴアドレス論理回路１７２に結合される。ＡＶＴベース・レジスタ１７４は、ＡＶＴ表全体のメモリにおける開始アドレスを含む。生起したアドレスを用いて、ＡＶＴアドレス論理回路１７２は、次にＡＶＴエントリ・レジスタ１８０の中に装填される、そのＡＶＴエントリに対するメモリをアクセスする。ＡＶＴアドレス論理回路１７２は、ＡＶＴ表に割り当てられたアドレス範囲内に入らないＡＶＴ頁数アドレスを検出するＡＶＴマスク・レジスタ１７５も含む。規則は、ＡＶＴマスク・レジスタ１７５のあるビット位置が０であるならば、ＡＶＴ頁数アドレスの対応ビットも０でなければならない；そうでなければ、マスク検査論理回路１７７がマスク誤りを検出しかつメモリ２８へのアクセスを否定すべく動作するというようなものである。ＡＶＴエントリ・アドレス生成及びマスク動作は、図１７によく示されている。図１７が図式で示すように、レジスタ・セグメント１７０ｃの２０ビットＡＶＴ頁数値の高位８ビット部分は、ＡＶＴ表エントリ・アドレスの高位部分（ビット１６－３１）を生成すべくＡＶＴベース・レジスタ１７４の内容と合計される。同時に、レジスタ・セグメント１７０ｃからのＡＶＴ頁数アドレスの残りの（下位）１２ビットは、ＡＶＴエントリ・アドレスの部分を直接形成する。ＡＶＴエントリは、４倍長語(quadword)量なので、それらは、４倍長語境界上に位置決めされる；それゆえに、ＡＶＴエントリ・アドレスの下位４ビットは、示すように、常にゼロである。

【００８５】図１７は、マスク動作も示す。ＡＶＴ頁数アドレスの高位２バイトは、マスク・レジスタ１７５に含まれるマスクと比較される。０を含んでいるマスク・レジスタのビット位置が“１”を有する高位２バイトの対応ビット位置を検出したならば、マスク・レジスタは、メモリ２８へのアクセスを否定する“マスク誤り（Ｍａｓｋ　Ｅｒｒｏｒ）”信号をアサートし、かつプロセッサ２０による作用に対して割込み論理回路８６（図９）への割込みを生成しかつポストする。マスク動作は、ＡＶＴエントリの表のサイズを変化させる。ＡＶＴマスク・レジスタ１７５の内容は、プロセッサ２０にアクセス可能であり、プロセッサ２０にＡＶＴエントリ表のサイズを随意に選択させる。最大ＡＶＴ表サイズは、あらゆる３２ビットＴＮｅｔアドレスの検証（及び変換）を許容する；即ち、最大サイズＡＶＴエントリ表は、２０^{20}の異なる頁アドレスを検証しかつ変換することができる。最小サイズＡＶＴ表は、あらゆる２４ビットＴＮｅｔアドレス（即ち、高位８ビットがゼロであるようなＴＮｅｔアドレス）の検証及び変換を許容する。最小ＡＶＴ表は、２０^{12}の異なる頁アドレスを検証しかつ変換することができる。

【００８６】従って、ＡＶＴ表エントリが１６バイトであるので、最大サイズＡＶＴ表は、専用メモリ空間の１６メガバイトを必要とする。しかしながら、ＡＶＴマスク・レジスタ１７５及びＡＶＴアドレス論理回路１７２の内容によって実行されるマスク動作は、ＡＶＴサイズをシステムのニーズにマッチさせる。非常に多数の外部素子（例えば、システム中のＩ／Ｏ装置の数は、大きい）を含む処理システム１０は、広範囲のＴＮｅｔアドレス、及び対応ＡＶＴエントリを必要とし、かつＡＶＴエントリにメモリ２８のかなりの量のメモリ空間を確保し（割り当て）なければならない。逆に、小さな数の外

部素子を有する、小さな処理システム１０は、小さなＡＶＴ表を用いることができようにそのような大きなＴＮｅｔアドレス要求を有さず、メモリ空間を節約する。従って、小さなシステムでは、高位ビットは、用いられない（または、より正確には、用いられるべきでない）。小さなＡＶＴ表が順番であるときには、ＴＮｅｔアドレスの高位ビットは、ゼロであるべきである；特定のシステムに対して範囲外であるＴＮｅｔアドレスを有するＡＶＴ表エントリをシークする試みは、誤りである。マスク・レジスタ１７５の内容を用いて、そのような誤りを検出することはマスク論理回路の機能である。それゆえに、あらゆるＣＰＵ１２（またはこの妥当性検査技術を用いている他のシステム素子）に対するときのＡＶＴ表拡張の許容可能サイズは、論理“ＯＮＥ（１）”にセットされるビット位置によりマスク・レジスタ１７５の内容によって示される。論理“ＺＥＲＯ（０）”にセットされるマスク・レジスタ１７５のビット位置は、処理システム１０の限界の外側の、存在しないＴＮｅｔアドレスを表わす。許容可能ＴＮｅｔ範囲外のＴＮｅｔアドレスを有する受信パケットは、それがらＺＥＲＯであるべきところの論理ＯＮＥにセットされたビット位置を有する。ＡＶＴアドレス論理回路１７２は、この範囲外ＴＮｅｔアドレスを検出し、かつＡＶＴ誤り割込みを発行させる。

【００８７】メモリ２８で保守されることが必要なＡＶＴ表のサイズを変化することができることに加えて、上述したように、図１７に示した技術は、また、可撓性を伴ってメモリ2.8にＡＶＴ表を位置決めさせるということが、いま当業者に明らかであろう。図１７は、ＡＶＴ表が 2^{17}（１２８Ｋ）の累乗境界上に位置決めすることができるということを示す。各ＡＶＴエントリは、妥当性検査処理の間中にＡＶＴエントリ・レジスタ１８０に保持されるときに図１６に示したフィールドを含む１２８ビット４倍長語である。ＡＶＴエントリは、二つの基本フォーマットを有する：標準及び割込み。標準ＡＶＴエントリのフォーマットは、図１８（及び、ＡＶＴエントリ・レジスタ１８０の内容を示すことによって、ある程度、図１６にも）に示される；割込みフォーマットは、図２０に示される。ＡＶＴ論理回路９０の説明を続ける前に、ＡＶＴエントリの意味及び内容の理解は、役に立つであろう。図１８を参照すると、標準ＡＶＴエントリが５２ビットＰｈｙｓｉｃａｌ Ｐａｇｅ Ｎｕｍｂｅｒ（物理頁数）フィールドを含んで示されている。このフィールドの内容は、その内でアクセスがメッセージ・パケットの要求ソースを許容されるメモリ２８の頁の物理アドレスを識別する。（一般に、各メモリ頁は、４Ｋ（４０９６）バイト・メモリ位置を含む。）Ｐｈｙｓｉｃａｌ Ｐａｇｅ Ｎｕｍｂｅｒフィールドの内容は、（妥当性検査をシークしているメッセージ・パケットのヘッダから引き出された）ＡＶＴ入力レジスタ１７０に保持される１２ビット・オフセット・フィールド１７０ｄの内容と連結される。結果は、妥当性検査が許可されるならば － データが書込まれるかまたは読取られるメモリ２８内の位置の物理アドレス全体である。

【００８８】アクセスが特定の４Ｋ頁の全メモリ位置に認められうると同時に、アクセスは、また、その頁の部分だけに制限されうる。後者の制限を実施するために、ＡＶＴエントリは、アクセスが許可されるメモリ２８の識別された頁内の上部及び下部バウンドを画定する二つの１２ビット・フィールド（上部バウンド、下部バウンド；図１８）を含む。特に、ＡＶＴエントリの下部バウンド・フィールドは、このＡＶＴ表エントリが適用される最低値を有するバイトのメモリ頁を有する（メモリ頁内の）オフセットを指定する。上部バウンド・フィールドは、このＡＶＴ表エントリが適用される最高アドレスを有するバイトのメモリ頁内のオフセットを指定する。この値（例えば、オフセット値１７０ｄ＋ＡＶＴ入力レジスタ１７０のＬｅｎフィールド１７０ｂの内容）を渡すメモリ位置をアクセスする試みは、割込みを介してプロセッサにポストされる誤りを結果として生ずる。１２ビット“Ｐｅｒｍｉｓｓｉｏｎｓ（許可）”フィールドは、ＡＶＴエントリに対応している要求ソースに認められた許可を指定すべくＡＶＴエントリに含まれる。Ｐｅｒｍｉｓｓｉｏｎｓフィールドは、図１９に示されており、あるＰｅｒｍｉｓｓｉｏｎｓサブ・フィールド（Ｅ，ＰＥＸ，ＰＥＹ，Ｉ，Ｃ，Ｗ，Ｒ，及びＢ）は、メモリ・アクセスへの以下の認定を識別する：

【００８９】Ｅ：（Ｅｒｒｏｒ Ｅｎａｂｌｅ（誤りイネーブル））このＡＶＴエントリを通って指向される誤ったアクセスは、このフィールドが二つの特定状態に一つ（例えば“ＯＮＥ”）にセットされるならば（割込み論理回路に）報告される。

【００９０】ＰＥＸ：（Ｐａｔｈ Ｅｎａｂｌｅ Ｘ（経路イネーブルＸ））この１ビット・フィールドの状態は、（全ての他の適用可能な許可も合致したならば）このＡＶＴエントリを用いるべくゼロに等しいヘッダの“ｐａｔｈ（経路）”ビットで受信したメッセージ・パケットをイネーブルすべく“ＯＮＥ”にセットされる。このビットが“ＺＥＲＯ”にセットされたならば、アクセスは、“ｘ経路”（経路＝０）にわたり受信したＡＶＴエントリが適用されるメッセージ・パケットが否定される。否定は、割込み論理回路で割込みとしてログされ、かつＥフィールドが誤り－報告をイネーブルする状態（“ＯＮＥ”）にセットされたならば－ プロセッサ２０に報告される。

【００９１】ＰＥＹ：（Ｐａｔｈ Ｅｎａｂｌｅ Ｙ（経路イネーブルＹ））この１ビット・フィールドは、それがＯＮＥ（１）にセットされた経路ビットで受信したメッセージ・パケットに適用されるということを除き、ＰＥＸフィールドと同じ方法で動作する。

【００９２】Ｉ：　（Ｉｎｔｅｒｒｕｐｔ（割込み））　このビットが（例えば、“ＯＮＥ”に）セットされたならば、他のフィールド（上部バウンド、等）は、割込み書込みを処理しかつ割込みキューを管理するための新しい定義を得る。これは、割込み論理回路８６の説明に関して以下により詳細に説明する。

【００９３】Ｃ：　（Ｃａｃｈｅ　Ｃｏｈｅｒｅｎｃｙ（キャッシュ・コヒーレンシー））これは、どのようにメモリ２８への書込み要求が処理されるかを指定すべく符号化された、２ビット・フィールドである。一つの状態にセットすると、要求書込み動作は、普通に処理される；第２の状態にセットすると、メモリのＡＶＴエントリ・マップド領域の上部または下部バウンドに含まれた部分キャッシュ回線(fractional cache line) を有するアドレスを指定している書込み要求は、以下に説明する、割込みハンドラ２５０（図２１）によって保守されるキャッシュ・コヒーレンシー・キューに書込まれる。これは、第３の状態にセットされた全キャッシュ回線アライメントを有さないメモリ２８のユーザ・データ構造またはバッファ領域への書込み転送をＣＰＵ１２に管理させ、このＡＶＴエントリをアクセスする全ての書込み要求は、キャッシュ・コヒーレンシー・キューに書込まれる。第４の状態にセットすると、このＡＶＴエントリによって参照される物理メモリ位置は、ハードウェア・コヒーレンシー機構を用いてアクセスされる。

【００９４】Ｗ：　（Ｗｒｉｔｅ　Ａｃｃｅｓｓ（書込みアクセス））　この１ビット・フィールドの状態は、Ｌｏｗｅｒ（下部）及びＵｐｐｅｒ（上部）Ｂｏｕｎｄ（バウンド）フィールドによって識別されるメモリ領域内で　－　要求ソースに対してメモリへの書込みアクセスを認めるかまたは否定する。

【００９５】Ｒ：　（Ｒｅａｄ　Ａｃｃｅｓｓ（読取りアクセス））　この１ビット・フィールドの状態は、指定したメモリ領域内で　－　要求ソースが読取り動作に対してメモリへのアクセスを有するか否かを決定する。

【００９６】Ｂ：　（Ｂａｒｒｉｅｒ　Ａｃｃｅｓｓ（バリア・アクセス））　この１ビット・フィールドの状態は、指定したメモリ領域内で　－　要求ソースがバイア動作（以下に説明）に対してメモリへのアクセスを有するか否かを決定する。そして、ＡＶＴエントリの２０ビット“ＳｏｕｒｃｅＩＤ”フィールドは、ＡＶＴエントリの許可情報が適用される特定のソースを識別する。図１６に示されたＡＶＴ論理回路にここで戻ると、一度ＡＶＴエントリのアドレスが形成されると、エントリは、アクセスされかつＡＶＴ表エントリ・レジスタ１８０に一時的に記憶される。ＡＶＴエントリ・レジスタ１８０に含まれるように、Ｐｅｒｍｉｓｓｉｏｎｓフィールドの内容は、アクセス論理回路１８４によりＡＶＴエントリ・レジスタに保持されるＴｙｐｅフィールドによって指定されたように、要求されるアクセスの型と比較される。要求されたアクセスが許可されたものと一致しないならば、アクセスは、否定され、あつアクセス論理回路１８４は、ＯＲゲート１８４及びＡＮＤゲート１８６を含んでいる誤り生成論理回路を介して生成されたＡＶＴエントリ割込み信号をもたらすべく誤り信号（“Ｎｏ”）をアサートする。シークされたアクセスの型がＰｅｒｍｉｓｓｉｏｎｓの一つでないならば、アクセスは、否定される。

【００９７】（ＡＶＴエントリ・レジスタ１８０の“ｓｒｃ　ＩＤ”値として識別される）アクセスされたＡＶＴエントリのＳｏｕｒｃｅＩＤフィールドは、用いられるＡＶＴエントリに対応するソースを指定し、かつ比較論理回路１９０によって要求メッセージ・パケットに含まれるＳｏｕｒｃｅＩＤと比較される。再び、ミスマッチは、ＡＶＴ　Ｅｒｒｏｒ　Ｉｎｔｅｒｒｕｐｔ（誤り割込み）を生成させることを比較論理回路１９０に結果として生じ、そしてアクセスが否定される。同時に、ＡＶＴエントリのＬｏｗｅｒ　Ｂｏｕｎｄフィールド（ＡＶＴエントリ・レジスタ１８０において“ｌｗｒ　ｂｎｄ”として図１６に示される）は、それがＡＶＴ入力レジスタ・セグメント１７０ｄにおけるＯｆｆｓｅｔ（オフセット）値と比較される論理回路１９４を比較すべく適用される。Ｏｆｆｓｅｔ値がＡＶＴエントリのＬｏｗｅｒ　Ｂｏｕｎｄフィールドに含まれたものよりも小さく、アクセスが許可された頁部分の外側であるということを示しているならば、コンパレータ（比較器）１９４は、ＯＲゲート１８４及びＡＮＤゲート１８６を介して、ＡＶＴ誤り割込みを生成する、信号を始動し、メモリ２８へのアクセスを否定する。

【００９８】同様に、比較論理回路１９６は、（書込み動作が要求されたならば）書込まれるべきデータの量が、（エントリのＬｏｗｅｒ及びＵｐｐｅｒ　Ｂｏｕｎｄフィールドによって画定されるように）要求ソースに割り当てるメモリ空間の量を超えるかどうかを決定すべく　－　加算器論理回路２００から　－　ＵｐｐｅｒＢｏｕｎｄフィールド（ＡＶＴエントリ・レジスタ１８０の“ｕｐｒ　ｂｎｄ”）をＬｅｎフィールド（即ち、書込まれるべきデータ・バイトの数）及びＯｆｆｓｅｔの合計と比較する。アクセス要求が確認されたならば、ＡＶＴエントリ・レジスタ１８０のＰｈｙｓｉｃａｌ　Ｐａｇｅ　Ｎｕｍｂｅｒ　（ｐｈｙｓ　ｐｇ　＃）内容は、アクセスが行われるメモリ位置をアドレス指定すべく、図１７に関して上述したように、ＡＶＴ入力レジスタ１７０からのＯｆｆｓｅｔと一緒に、用いられる。システム１０の素子（例えば、装置１７とＣＰＵ１２；図１、図２及び図３）間のメッセージ・パケットの通信は、ことのほか、アクティビティを要求するか、またはアクティビティを報告するか、或いは誤りの発生を知らせるべく割込みを分配する、新規な方法にも採り入れられる。それゆえに、割込みメッセージ・パケット送付

は、他の素子間通信と同じ方法でＴＮｅｔネットワーク・システムを用い、かつ３つのステージ進行を含む：（１）ソース素子からの割込みメッセージ・パケットの生成及びタスク指名（ディスパッチ）；（２）その宛先へのＴＮｅｔネットワークを通る割込みメッセージ・パケットの伝播；（３）宛先での作用に対する解釈及び“ポスティング”。あらゆるシステム素子は、割込みメッセージ・パケットの受信者でありうる。宛先がＣＰＵであるならば、割込みメッセージ・パケットは、実質的に、メッセージ・パケット・ヘッダのＤｅｓｔｉｎａｔｉｏｎ　ＩＤフィールド（図５（ｂ））がＣＰＵを識別し、かつＡｄｄｒｅｓｓフィールドがどのように割込みメッセージ・パケットが処理されるべきかの命令を含んでいるＡＶＴエントリ（Ｉｎｔｅｒｒｕｐｔ　Ｄｅｓｃｒｉｐｔｏｒ）を選択するような標準“書込み”要求である。

【００９９】割込みアクティビティの始動するための許可は、また、ＡＶＴ論理回路８８によって確認されなければならない。それゆえに、受信したメッセージ・パケットは、割込みを説明する割込みデータを含む。その割込みデータは、割込みが受信されかつ“ポスト”され、そしてプロセッサ２０によってサービスする準備ができていることを示すべくプロセッサ２０に供給する信号で、メモリ２８の特定のキュー（割込みキュー）に書込まれるべきである。割込みキューは、特定のメモリ位置にあるので、プロセッサは、必要なときに割込みデータを得ることができる。割込みに対するＡＶＴ割込みエントリは、二つの型の一つでありうる：マルチエントリ・キュー型割込み、または単一エントリ・キュー型割込み。ＡＶＴ割込みエントリの両方の型に対するフォーマットは、基本的に同じであり、そのフォーマットは、図２０に示されている。マルチエントリ・キュー型割込みに対するＡＶＴ割込みエントリは、割込みを送付すべく最初に構成されたか、またはルータ１４或いは受信ＣＰＵ（例えば、不良ＣＲＣ）によって検出された例外により途中で割込みになったかのいずれかである受信したメッセージ・パケットに対して用いられる。これらのエントリは、上述したのとほぼ同じ方法でメッセージ・パケットを確認すべくＡＶＴ論理回路９０によって用いられ、かつ割込みメッセージ・パケットのヘッダ、及び付随しているデータが記憶されるメモリ２８の円形キューを識別すべく割込み論理回路８６（図９及び図２１）によって用いられる。更に、割込み論理回路８６は、信号受信及び／又はマルチエントリ割込みの生成に対して割込みまたは“原因”レジスタ２８０（図２１：以下により完全に説明される）にビットをセットする。

【０１００】単一エントリ・キュー型割込みに対するＡＶＴ割込みエントリは、ＡＶＴ割込みエントリが記憶に対してメッセージ・パケット情報を指向する割込みデータ構造が、メモリ２８における（変更可能であるが）固定された位置であるということを除き、実質的に同じような方法で動作する。両方のＡＶＴ割込みエントリ型（マルチエントリ及び単一エントリ割込み）は、図２０に示した４倍長語（１２８ビット）フォーマットを有する。ＡＶＴ割込みエントリの６４ビット・セグメント（“Ｑｕｅｕｅ　Ｂａｓｅ　Ａｄｄｒ（キュー・ベース加算器）”）は、割込みデータが書込まれる割込みキューのメモリ２８の位置へのポインタとして用いられる。割込みキューは、割込みをサービスするときに、割込みデータが割込み論理回路８６によりＦＩＦＯキューのテール(tail)で受信されかつ挿入され、かつプロセッサ２０によりキューのヘッド(head)から抽出されるようにＦＩＦＯの形で構成される。また、ＡＶＴ割込みエントリは、ソースＩＤ情報を含んでいる２０ビット・セグメント（“ＳｏｕｒｃｅＩＤ”）も含み、割込み処理による注意をシークしている外部装置を識別する。ＡＶＴ割込みエントリのソースＩＤ情報が、コンパレータ１９０（図１６）によって実行される比較によって決定されるように、入力メッセージ・パケット（Ｓｏｕｒｃｅ；図５（ｂ））のヘッダに含まれたものとマッチしないならば、割込みキューへのアクセスは、否定され、かつＡＶＴ誤り割込みが生成される。

【０１０１】ＡＶＴ割込みエントリの１２ビット“Ｐｅｒｍｉｓｓｉｏｎｓ”セグメントは、標準ＡＶＴエントリに関して上述したのと同じ許可情報を含む。しかしながら、一般に、割込みを送付するメッセージ・パケットは、書込み要求として構成され、それがメモリ２８に運ぶ割込みデータを書込むべくシークする。それゆえに、適切なＡＶＴ割込みエントリは、割込みデータをメモリ２８に書込ませるべくセットされたＷｒｉｔｅ　Ａｃｃｅｓｓ（書込みアクセス）ビット（Ｗ）を有する。Ｐｅｒｍｉｓｓｉｏｎｓフィールドの割込みビット（Ｉ）は、セットされたときに、割込みメッセージ・パケットを確認しかつ処理するためのものとしてＡＶＴ割込みエントリを識別する。そして、ＡＶＴ割込みエントリの４つの、１バイト・セグメント（“ｃ”，“ｑ”，“ｌ”，及び“ｚ”）は、（１）プロセッサ２０にセットされた割込みレベルを決定するために用いられる割込みの“ｃｌａｓｓ（クラス）”（以下により完全に説明される）；（２）、その内容が（Ｑｕｅｕｅ　Ｂａｓｅ　Ａｄｄｒｅｓｓ（キュー・ベース・アドレス）フィールドによって識別された）特定のキューのどこに割込みデータが書込まれるべきであるかということを示す、レジスタを、分かるように、選択するために用いられるキュー数；（３）そこに記憶することができる倍長語の数による各キュー位置で利用可能な記憶装置のサイズまたは量；及び（４）キューのどこにデータが書込まれるかを識別するために用いられるキュー・テール・カウンタにおけるビットの数、をそれぞれ識別する。Ｑｕｅｕｅ　Ｂａｓｅ　Ａｄｄｒ、とｃ，ｑ，ｌ，及びｚセグメン

トは、メモリ２８の位置をポイントすべく割込み論理回路８６によって用いられる。割込み論理回路８６は、それぞれが割込みデータを挿入することができる４つのキューの一つをポイントする４つの“テール”・カウンタを含む。４つのカウンタの特定の一つは、ＡＶＴ割込みエントリのｑセグメントの内容によって選択される。そこから割込みエントリが引き出される点である、キューの他端は、４つの“ヘッド”・カウンタの一つによって識別される。ヘッド及びテール・カウンタの（ビット数による）サイズは、以下に示す表２に指定されたように、９により負にバイアスされた、ｚサブ・フィールドによって指定される。キュー・テール・カウンタ・サイズは、テール・ポインタがゼロの値にラップバック(wrap back）するときを決定するために用いられる。各エントリのサイズによって分割された語の数（バイト）は、キュー・エントリの数を与える。最短キューは、ほんの３２エントリ（１２８バイト・エントリによって分割された４ｋＢキュー）を有して、４ｋＢを必要とすると同時に、最長キューは、３２，７６８エントリ（エントリ当たり１６バイトによって分割された５１２ｋＢ）と同じ数だけ有することができる。

【０１０２】

【表２】表　２

Ｚ	解　　釈
０	５１２　倍長語　（４Ｋ　バイト）
１	１Ｋ　倍長語　（８Ｋ　バイト）
２	２Ｋ　倍長語　（１６Ｋ　バイト）
３	４Ｋ　倍長語　（３２Ｋ　バイト）
４	８Ｋ　倍長語　（６４Ｋ　バイト）
５	１６Ｋ　倍長語　（１２８Ｋ　バイト）
６	３２Ｋ　倍長語　（２５６Ｋ　バイト）
７	６４Ｋ　倍長語　（５１２Ｋ　バイト）

８－１５　使用せず

【０１０３】割込みキューの各割込みエントリのサイズは、以下の表３に示した方法で、４倍長語で、１フィールドによって指定される。

【０１０４】

【表３】表　３

１	解　　釈
０	１　４倍長語　（１６　バイト）
１	２　４倍長語　（３２　バイト）
２	４　４倍長語　（６４　バイト）
３	８　４倍長語　（１２８　バイト）

４－１５　使用せず

【０１０５】－　割込み処理：上記したように、本発明の新規の特徴（機能）は、サービスをするためにＣＰＵ１２に割込みを送付すべくＴＮｅｔネットワーク・メッセージ送付能力を用いる能力（機能）である。例えば、Ｉ／Ｏ装置は、トランザクションを送付したメッセージ・パケットにおける不適当なアドレス、またはＣＲＣ誤りを有するメッセージ・パケットの受信、または受信メッセージ・パケットが受信者を識別しなかった宛先アドレスを有していたということに気付く（注目する）ことのような多数の理由によりＣＰＵによって発行される読取りまたは書込みトランザクションを終了することができないであろう。Ｉ／Ｏ装置、またＩ／Ｏインターフェイス素子によって気付かれた（注目された）、これら及び他の誤り、例外、及び不規則性は、ＣＰＵの介入を必要とする条件（状態）になることができる。従来のシステムでは、そのような条件（状態）は、割込みの対象事項である；そこで、それらは、そのような割込みが送付されないということを除き、過去におけるように　－　割込み条件（状態）についての情報を殆どまたは全く有さなず、かつそのような目的のためにだけ確保された信号回線によって　－　しかしシステムのＩ／Ｏ素子に利用可能なメッセージ・システムを通して、ここにある。この特徴（機能）は、余分な信号回線に対する必要性を低減する（他の使用に対して信号回線空間が利用可能であるものを許容する）だけでなく、ＣＰＵが調査を実行するために時間を費やさなくてもよいように割込みの原因になったものに対するより多くの情報を供給することができるファシリティを供給する。

【０１０６】この特徴（機能）により、メモリ２８に書込まれるべき割込みタスク指名を含んでいる、入力メッセージ・パケットは、妥当性検査のためにＡＶＴ論理回路９０（図１６）にまず渡される。また、ＡＶＴ論理回路９０は、メッセージ・パケットが正規のＩ／Ｏ書込み要求、割込み、あるいは禁止されているメモリ２８への誤アクセスであるか否かを決定する。ＡＶＴ論理回路９０のＡＶＴベース・レジスタ１７４の内容は、標準メッセージ・パケットに対して上述したのと同じような方法で主メモリにおけるＡＶＴ割込みエントリに対するポインタを生成するためにＡＶＴ入力レジスタ１７０に含まれる頁数フィールド１７０ｃ（図１６）を伴って用いられる（例えば、データを読取るかまたは書込むためにメモリ２８へのアクセスをシークすること）。形成されたアドレスによってそのように識別されたＡＶＴエントリは、メモリ２８からアクセスされかつインターフェイス装置２４（図９）の割込み論理回路８６による使用のためにＡＶＴエントリ・レジスタ１８０にセットされ、図２１に詳細に示される。一度その割込み情報を運んでいるメッセージ・パケットがＡＶＴ論理回路９０によってクリアされたならば、割込み情報を処理する役割を果たすのは、割込み論理回路８６である。割込み論理回路８６は、４つのキュー・テール・レジスタ２５６の内容を受信し、かつそれらの間で選択する、マルチプレクサ（ＭＵＸ）２５２を含んで、図２１に示されている。同様に、ＭＵＸ２５４は、４つのキュー・ヘッド・レジスタ２６２の内容を受信し、かつそれらの間で選択する。各ＭＵＸ２５２，２５４の選択入力は、ＡＶＴエントリ

・レジスタ１８０に保持された（割込みメッセージ・パケットに対応している）検索ＡＶＴエントリの“ｑ”セグメントの内容を受信すべく結合される。使用すべくキュー・レジスタ２５６，２６２の各グループからの一つを選択するために用いられるのは、ｑ値である。

【０１０７】数は、ここでは、以下に説明する理由により４つに制限されるが、割込みデータの記憶を処理するためにメモリにセット・アップされたあらゆる数のキューが存在しうる。各そのようなキューのメモリ２８内の位置は、アクセスされたＡＶＴエントリのキュー・ベース・アドレス値によって指定され、かつエントリ・レジスタ１８０（図１６の“ｐｈｙｓ　ｐｇ＃”；　図２１の“ベース”）に保持される。４つのキュー・テール・レジスタ２５６の内容は、それぞれが特定のキューの中にオフセットを形成してキュー・ベース・アドレス値によって指定される。選択したキュー・テール・レジスタ２５６の内容は、加算器２５８によりキュー・ベース・アドレスと組合わされ、そこで割込みデータが書込まれる指定されたキューの中にエントリ・ポイントを形成する。さらに多くのまたは少ないキューを維持（保守）することができるということは、当業者には、明らかなことであるが、４つのキュー・ヘッド及びテール・レジスタ２６２，２５６は、４つのキューを処理することだけに割込み論理回路８６を制限する。レジスタ２５６は、特定のキューの“テール”の位置を指定し、次の受信割込みデータが配置されるキュー・エントリを指し示す。４つのキュー・ヘッド・レジスタ２６２は、特定のキューの他端を指定する。

【０１０８】キュー・ベース・アドレスと選択されたテール・キュー・レジスタ２５６の内容との組合せから生起されたアドレスは、４倍長語（１６バイト）境界上に位置合わせすべく形成されるのが好ましい。これは、キュー・エントリ・アドレスの下位４ビットを０に強要することによって達成される。キュー・エントリ・アドレスの形成は、図２２に図式的に示され、選択されたテール・レジスタ２５６の１５ビット内容の高位７ビットが、ＡＶＴエントリ・レジスタ１８０に含まれるキュー・ベース・アドレス・フィールドのビット位置１２－３１に付加されることを示している；この合計の結果は、キュー・エントリ・アドレスの高位２０ビット（ビット位置１２－３１）を形成する。選択されたテール・レジスタ２５６の下位８ビット内容は、キュー・エントリ・アドレスの下位ビット位置４－１１として直接用いられる。上述したように、キュー・エントリ・アドレスの下位４ビット（ビット位置０－３）は、所望の位置合わせに対して全てゼロに強要される。纏めると、割込みを含んでいるメッセージ・パケットは、それらが、実質的に、受信ＣＰＵ１２のメモリ２８にデータを書込むための要求であるので、他のメッセージ・パケットと同じ方法で最初処理され、その要求は、ＡＶＴ論理回路９０によって確認されなければならない。それゆえに、メッセージ・パケットからの情報は、ＡＶＴ入力レジスタ１７０及びＡＶＴエントリを位置決めしかつメモリ２８からアクセスするために用いられる部分（フィールド１７０ｃ及び１７０ｄ）にセットされる。ＡＶＴエントリは、メッセージ・パケットが割込み処理に対する適当な情報を含むならば、ＡＶＴエントリ・レジスタ１８０にセットされる割込みＡＶＴエントリでありかつ割込みを確認（オーセンチケート）するために用いられ、そして、割込み論理回路８６を用いて、ＡＶＴエントリに含まれるベース・アドレス情報によって指定された４つの円形キューの一つに割込みデータを記憶する。プロセッサ２０は、通知され、かつ割込みが処理されるか否か、及びどのように処理されるということは、それら次第である。

【０１０９】キューに割込みメッセージ・パケット・データを記憶して、アドレス指定は、次のメッセージ・パケットの割込みデータの受信を見越して更新されなければならない。割込みデータが選択キューに書込まれた後、ＡＶＴ表エントリ・レジスタ１８０に含まれる“１”フィールドの内容は、組合せ(combiner)回路２７０により選択テール・キュー・レジスタ２５６と組合わされ、その出力は、次の割込みメッセージ・パケットの割込みデータが記憶されるところで新しいオフセットをキューに変える(turn into) べく“ｍｏｄ　ｚ”回路２７３によって処理される。その新しいオフセット値は、選択テール・キュー・レジスタ２５６に戻される。同時に、組合せ回路２７０の出力は、比較回路２７２に供給される。割込み問合わせは、ｚのモジュール・サイズを有するファッションで環状であるべく構成される。ｍｏｄ　ｚ回路は、環状性を維持する出力を生成する。テール・キュー・ポインタがキューにおけるネスト・エントリ・ポイントを識別し、かつヘッド・ポインタがキューにどのくらいのルームが残っているのか、対応テール・ポインタに関して、識別するので、これら二つの値が等しいならば、キューは、充満（いっぱい）である。それゆえに、（選択されたヘッド・キュー・レジスタ２６２によって供給された）ヘッド・ポインタを最後のエントリの結果として生成されたテール・ポインタと比較することによってこの決定を行うのは、比較回路２７２である。キューに対してヘッド及びテール・ポントがいま等しいならば、比較回路２７２は、それ自身が割込み信号である、“Ｑｕｅｕｅ　Ｆｕｌｌ（キュー充満）”警告信号を発行する。Ｑｕｅｕｅ　Ｆｕｌｌ警告信号は、事項(matter)が適格に処理されないならば、キューがいっぱいであったならば、追加割込みメッセージが廃棄されるので、遅く受信した割込みデータが失われうるという、警告としてプロセッサ装置２０に運ばれる“固有(intrinsic)”の割込みになる。

【０１１０】入力メッセージ・パケット割込みは、割込みレジスタ２８０の多数のビット位置の一つをまずセッ

トすることによって割込みをプロセッサ２０にポストさせる。マルチーエントリ・キュー型割込みは、プロセッサ２０にポストするために割込みレジスタ２８０ａにセットされる；単一エントリ・キュー割込みは、割込みレジスタ２８０ｂを用いる。どのビットがセットされたかは、ＡＶＴ入力レジスタ１８０に保持されたＡＶＴエントリのクラス・フィールド（ｃ）に依存する。

【０１１１】まずマルチーエントリ・キュー型割込みを考慮すると、マルチーエントリ・キュー型割込みが決定された後すぐに、インターフェイス装置は、復号回路２８３に印加（供給）される対応割込み信号（Ｉ１）をアサートする。復号回路２８３は、セットすべきレジスタ２８０ａのレジスタ位置を決定すべくＡＶＴエントリ・レジスタ１８０からクラス（ｃ）値を受信しかつ復号し、それによりプロセッサ２０に受信割込みに関する先進の情報、即ち、（１）ポストされた割込みの型、及び（２）その割込みのクラス、を供給する，同様に、単一エントリ・キュー割込みは、受信したときに、セットすべきレジスタ２８０ｂのビット部分を決定すべくクラス（ｃ）値を受信しかつ復号する、復号論理回路２８７に対応割込み信号（Ｉ２）をアサートさせかつ印加させる。

【０１１２】テール及びヘッド・キュー・レジスタ２５６，２６２は、マルチプレクサ（ＭＵＸｓ）２７６，２７４の別のペアにも結合される。更新レジスタ２７８の内容は、比較回路２７９によって互いに比較されるレジスタ２５６，２６２の対応ペアを選択する。更新レジスタは、比較のためのレジスタ・ペアを選択するためにプロセッサ２０によって書込み可能である。二つの選択されたレジスタ２５６，２６２の内容が等しく、対応キューが空であるということを示していることがわかったならば、対応割込みレジスタは、クリアされる。クラス・レジスタ２８１は、クリアされることが必要な割込みレジスタ２８０ａの割込みビットを（クラスにより）選択する。ちょっと脇道にそれると、プロセッサ２０に係わる二つの割込みの基本型が存在する：メッセージ・パケットによりＣＰＵ１２に伝達される割込みと、“固有”割込みと呼ばれる、ＣＰＵ１２自身によって生成される割込みである。固有割込みは、割込み論理回路８６の比較回路２７２によって生成されるキュー充満警告信号のような内部的に検出された誤りの結果として生ずる。しかしながら、それらは、最初に割込みパケットとして送られなかったメッセージ・パケットを受信するときに気付く例外も含みうる。そのようなメッセージ・パケット割込みは、メッセージ・パケットが不良指令記号を有しているものとして検出されたこと、または受信メッセージ・パケットが不要ＣＲＣを有する（かまたは以下に説明するように、ＴＰＢ識別子でタグされる）ことを見出した結果として生ずる誤り。これらの固有割込みは、固有割込みがマルチーエントリ及び単一エントリ割込みが割込みレジスタ１８０ａ，１８０ｂのビット位置をセットすることによってポストされるのと同じ方法でポストされるところに固有レジスタ２８０ｃの特定ビット位置をもたらす。更に、メモリ２８に維持されるＡＶＴ表は、固有ＡＶＴ割込みのために確保された第１の数のエントリを有する。固有割込み信号が固有割込みレジスタ１８０ｃをセットすべく生成されたときには、それは、割込みをアクセスさせかつＡＶＴ論理回路９０のＡＶＴエントリ・レジスタ１８０に装填させた例外に対応しているＡＶＴエントリをもたらす。それ以後、割込みは、メッセージ・パケット送信型割込みとして同じ方法で処理される。

【０１１３】ビット毎に基づき、割込みレジスタ２８０ａ，２８０ｂ，及び２８０ｃの各々と関連するのは、それぞれ対応マスク・レジスタ２８２ａ，２８２ｂ，及び２８２ｃである。割込みレジスタ２８０（例えば、２８０ａ）の各ビット位置は、マスク・レジスタ２８２（例えば、２８２ａ）における対応ビット位置を有する。マスク・レジスタ２８２の特定ビットがセットされたときには、関連割込みの認識は、抑制される。割込みレジスタ２８０の内容は、マスク・レジスタ２８２の内容によって渡されたならば、それらが７つの割込み“ポスティング(postings)”（信号）に組合わされる、複数のＯＲゲートを含んでいる、組合せ論理回路２８６に結合される。組合せ論理回路２８６は、７つの割込みポスティングをラッチ２８８に結合し、ラッチからそれらは、ポスティングを受信しかつ保持するポインタ割込みレジスタを有するプロセッサ２０（２０ａ，及び２０ｂ）に結合される。加えて、レジスタ２８８の内容は、比較回路２８９に印加（供給）され、かつレジスタ２８８の入力と（レジスタ２８８を装填する各クロックの前に）比較される。割込みにおける変化（割込みがサービスされ、かつそのポスティングがプロセッサ２０によって削除されたか、または新しい割込みがポストされたかのじずれか）を示している、相違が存在するならば、“ＣＨＡＮＧＥ（変化）”信号が、割込みポスティング変化が発生し、かつ変化をプロセッサ２０に伝達すべきであることをそれに報告すべくプロセッサ・インターフェイス６０に発行される。

【０１１４】ＡＶＴエントリ・レジスタ１８０は、ＴＡＧ及び有効ビットを備えた単一回線キャッシュのようの動作すべく構成されるのが好ましい。ＴＡＧは、システム・メモリ２８からＡＶＴエントリを調べる（ルックアップ）ために用いられるＴＮｅｔアドレスの部分からなる。通常動作では、ＴＡＧが入力パケットのＴＮｅｔアドレスとマッチしないならば、正しいＡＶＴエントリがシステム・メモリ２８から読取られかつＡＶＴエントリ・レジスタ２０６に読取られて、古いＡＶＴエントリを置換する。当業者は、ちょっと挙げてみると、セットアソシエイティブ、完全一関連(fully-associate)、また

は直接一マップ型(direct-mapped) のような他のキャッシュ編成が可能であるということを認識するであろう。

－ コヒーレンシー：キャッシュ・メモリ用いるデータ処理システムは、コヒーレンシーの問題を長く認識していた：キャッシュまたは主メモリへのアクセスが新鮮でないデータを決して戻さないか、または良好（最新）なデータを上書きしないことを確実にすること。この問題に対する多数の解決策が存在するが、その多くが大規模かつ複雑なハードウェアを使用する。コヒーレンシー問題は、データが外部（ＣＰＵに対して）Ｉ／Ｏまたは別のＣＰＵ１２からメモリに書込まれるとき、または、システム１０のコンテキストにおけるように（例えば、図４）、データがＣＰＵ１２ＢによってＣＰＵ１２Ａのメモリ２８に書込まれるときにも生ずる。一つの解決策は、入力データがバッファのバウンドがキャッシュ・ブロック境界に位置合わせされるようにメモリ・バッファに書込まれることを確実にすることである。しかしながら、この解決策は、入力データに対して用いられるキャッシュ・ブロックの妥当性を検査するためにソフトウェア・スキームと一緒に用いられるときにだけ応用（アプリケーション）を見出し、出力データに用いるキャッシュ・ブロックの書戻しを強要する。

【０１１５】それゆえに、（Ｉ／Ｏ、または別のＣＰＵ１２からの）入力読取り要求、及び出力読取り及び書込み要求に適するコヒーレンシー問題のソフトウェア管理に対する従来技術が存在する。しかしながら、従来技術は、キャッシュ・ブロック境界上に位置合わせされていないメモリ２８のＩ／Ｏバッファへの入力書込み要求を管理するのに向いていない。しかしながら、キャッシュ・ブロック境界上にメモリのＩ／Ｏバッファの位置合わせを要求することは、より可撓性が少ないシステム、及び既存の（オペレーティング・システム）ソフトウェアと非互換性でありうるシステムを結果として生ずる。従って、本発明の割込み機構は、そのバッファの境界がキャッシュ・ブロックの境界に位置合わせされるか否かということを考慮することなくデータ・バッファをメモリに位置決めさせるようにコヒーレンシーを確立するために用いられる。この関連において、入力パケットのソースがアクセスを許容されるメモリ２８の領域の上部及び下部境界（ｕｐｒ　ｂｎｄ，　ｌｗｒ　ｂｎｄ）を画定しているＡＶＴ表Ｅｎｔｒｙレジスタ１８０のフィールド（図１６）は、境界クロッシング（Ｂｄｒｙ　Ｘｉｎｇ）検査装置２１９に印加（供給）される。境界検査装置２１９は、それでＣＰＵ１２が動作すべく構成されるキャッシュ・ブロックのサイズの表示、ＡＶＴ　Ｅｎｔｒｙレジスタ１８０に保持されるＡＶＴエントリのＰｅｒｍｉｓｓｉｏｎｓフィールドからのコヒーレンシー・ビット（“ｃ［１：０］”）、及びＡＶＴ入力レジスタ１７０からのヘッダ情報のＬｅｎフィールドも受信する。Ｂｄｒｙ　Ｘｉｎｇ装置は、入力パケットのデータが　上に位置合わせされていないかを決定し、かつコヒーレンシー・ビット（“ｃ［１：０］”）が適切にセットされたならば、データ及びそのデータを含んでいるパケットのヘッダを記憶するために特別のコヒーレンシー・キューを指し示すべく用いられる割込みエントリのアドレスのフェッチを強要する。

【０１１６】図４２をここでちょっと参照すると、ＣＰＵ１２のメモリ２８（図４）によって実施されるメモリ空間の部分２８’が示されている。図４２が更に示すように、３つのキャッシュ境界CB_a，CB_b，及びCB_cは、メモリ部分２８’と一緒に含まれ、二つのキャッシュ・ブロックＣ＿BLK_a及びＣ＿BLK_bを画定する。書込み要求メッセージ・パケットが受信され（例えば、別のＣＰＵ１２、またはＩ／Ｏ装置から）、かつクロスーハッチング（格子縞）で示される、そのメッセージ・パケットに含まれるデータがメモリ部分２８’を含むメモリ２８の領域に書込まれるものであると想定する。事実、書込まれるべきデータは、キャッシュ・ブロックＣ＿BLK_aをほんの部分的に上書きするが、キャッシュ・ブロックＣ＿BLK_b、及び他のキャッシュ・ブロックを完全に上書きする。書込まれるＣＰＵ１２のキャッシュ２２がキャッシュ・ブロックＣ＿BLK_b、またはキャッシュ・ブロックＣ＿BLK_a以外の他のキャッシュ・ブロック（または、キャッシュ境界上に位置合わせされていないならば、入力データの他端を含んでいるキャッシュ・ブロック）を含むならば、ブロックは、“無効”とマークすることができ、メモリに書戻されかつ新しく受信したデータを上書きすることからそれを防ぐ。

【０１１７】しかしながら、キャッシュ２２がキャッシュ・ブロックＣ＿BLK_aを含むならば、ＡＶＴ９０の（“ｃ”がＰｅｒｍｉｓｓｉｏｎｓフィールドにセットされることによりイネーブルされたならば；図１６及び図１９を参照）境界クロッシング論理回路２１９は、キャッシュ・エントリを部分的に無効にするＩ／Ｏパケットを検出し、かつコヒーレンシー割込みを強要することが必要である。これは、特別割込みキャッシュへのポインタを含んでいる、割込み記述子のフェッチを結果として生じ、かつ全入力ＴＮｅｔ要求パケットは、キューに書込まれる。同時に、割込みは、入力データの一部分が特別のキューに装填されるということをプロセッサ２０に知らせるためにキュー型割込みレジスタ２８０に書込まれる。

【０１１８】要するに、入力パケットがメモリ２８に書込まれるべきデータを有するならば、境界クロッシング論理回路２１９は、データが書込まれるバッファの境界がキャッシュ境界と位置合わせされるかどうかを見る（調べる）べく検査する。そうであるならば、データは、指示（指導）されるように書込まれる。そうでなければ、パケット（ヘッダ及びデータの両方）は、特別の

キューに書込まれ、かつプロセッサは、上述した固有割込み処理によってそのように知らされる。次に、プロセッサは、特別のキューからキャッシュ２２にデータを移動し、よいデータが上書きされないかそもなくば失われず、かつキャッシュ２２とメモリ２８の間のコヒーレンシーが保存されることを確実にすべく後でキャッシュをメモリ２８に書込みうる。

【０１１９】－　ブロック転送エンジン（ＢＴＥ）：プロセッサ２０は、ＣＰＵ１２Ａの外部素子に情報を直接伝達（例えば、送る）することを禁止されているので、ＣＰＵのインターフェイス２４ａ（図９）のＢＴＥ８８は、情報送信の間接方法に対して供給される。ＢＴＥ８８は、情報のブロックを転送すべく全プロセッサ始動型（all processor initiated）Ｉ／Ｏトラフィックを実施するために用いる機構である。ＢＴＥ８８は、ＴＮｅｔパケット規定によった許容される最大値、現在は６４バイト、までの長さを有する読取り及び書込みパケットの生成を許容する。ＢＴＥ８８は、一つが他よりも高い優先度を与えられる、二つの“仮想”チャネルを供給する。図２３を参照する、ＢＴＥ８８は、それらの内容が（インターフェイス装置２４ａの；図９）ＭＵＸ３０６に結合されかつメモリ・コントローラ２６（図２３に図示せず）を介してシステム・メモリ２８をアクセスするために用いられる二つのＢＴＥレジスタ３００，３０２を含んで示されている。レジスタ３００，３０２の部分は、ＣＰＵ１２Ａ（図４）のメモリ２８におけるＢＴＥデータ構造３０４の始まりへのポインタ（例えば、ＢＴＥアドレス－０及びＢＴＥアドレス１）を含む。プロセッサ２０は、ＣＰＵ１２Ａの外部の一つまたは別の素子（例えば、ＣＰＵ１２ＢまたはＩ／Ｏパケット・インターフェイス１６のＩ／Ｏ装置１７，１８のどれか）に情報が送られるかまたはそれから情報が検索される度にメモリ２８にデータ構造３０４を書込む。各データ構造は、４倍長語境界上で始まることを必要とし、かつＢＴＥレジスタ３００，３０２は、プロセッサ２０によってのみ書込み可能である。プロセッサがＢＴＥレジスタ３００，３０２の一つに書込むときには、それは、ＢＴＥ状態マシン３０７によって制御される、ＢＴＥ処理を始動すべく動作する、クリア状態に要求ビット（ｒｃ０，ｒｃ１）をセットする語でそのようにする。

【０１２０】ＢＴＥレジスタ３００，３０２は、タイムーアウト／ＮＡＫ誤り表示を報告する誤りビット（ｅ０，ｅ１）も含む。誤りビットは、対応ＢＴＥレジスタが書込まれるときにクリアされる。誤り原因（ｅｃ）ビットは、タイムーアウトとＮＡＫｓとを差分する。情報が外部装置にプロセッサ２０によって転送されるときに、データ構造３０４のデータ・バッファ部分３０４ｂは、転送されるべき情報を保持する。外部装置からの情報がプロセッサ２０によって受信されたときには、データ・バッファ部分３０４ｂは、読取り応答情報を保持すべくターゲットされた位置である。データ構造３０４の始まり、プロセッサ２０によって書込まれた部分３０４ａは、送られるパケットを受信する外部素子を識別する、情報フィールド（Ｄｅｓｔ）を含む。部分３０４ａは、所望の動作（例えば、読取りまたは書込み情報）を記述する情報フィールド（ＴＹＰＥ）、書込まれるかまたは要求されるデータのバイト数を記載している、長さ情報フィールド（Ｌｅｎ）、及び所望のデータが位置決めされる外部素子（Ｄｅｓｔ）の場所、または送信されたデータが書込まれるべき場所を識別している、アドレス情報フィールド（Ａｄｄｒｅｓｓ）も含む。この情報は、図５（ａ）～図５（ｄ）及び図６～図８に示す形にパケットをアセンブルすべくパケット送信機装置１２０（図９）によって用いられる。

【０１２１】データ構造部分３０４ａにおけるアドレス情報にすぐ続くのは、データ・バッファ部分３０４ｂが位置決めされるメモリのアドレスを含んでいる語（Ｌｏｃａｌ　Ｂｕｆｆｅｒ　Ｐｔｒ）である。次に、チェイン・ポインタ、要求が終了したことを示すエンドーオブーリスト（ｅｌ）フラグ、終了インディケータ上の割込み（ｉｃ）及びチェックサム（ｃｓ）要求を含んでいる語が、その後にすぐ続く。一つのデータ構造３０４は、最大６４バッファ長まで外部素子（例えば、Ｉ／Ｏ記憶装置）に移動されるべくメモリのデータの各部分に対して用いられる。ＢＴＥは、データの６４バイト・セグメントに対して、各要求構造に応じて、メモリ２８を逐次的にアクセスすべく動作し、各セグメントに対してメッセージ・パケットを形成し、かつ進行してそのメッセージ・パケットを送る。チェイン・ポインタは、エンドーオブーリスト・ビット（ｅｌ）がセットされない限り、データの別の６４バイトに対する次のデータ構造にＢＴＥを指向して、動作を終わらせる。データが多数の外部素子へ送られるべきであるならば、各異なる素子は、セットアップされるべくそれ自身のデータ構造（６４バイト以上が送られるならば複数のデータ構造）に要求する。次に、これら個別のデータ構造は、要求構造のチェイン・ポインタ・フィールドに含まれるチェイン・ポインタを用いて、チェインで繋がれうる。チェイン・ポインタ・フィールドは、後続データ構造に対するＢＴＥレジスタの内容として用いられる。例えば、メモリ２８の大きなブロックのデータがＮの異なる外部素子に送られるべきであるならば、データ構造は、ＢＴＥ論理回路８８が送られるべきデータを見出すことができるメモリ２８の場所を識別している各データ構造でＮの外部素子のそれぞれに対するメモリに書込まれる。各素子に対するデータは、ＢＴＥ論理回路８８によってアクセスされ、メッセージ・パケットがデータを含んで形成され、かつそれらが適切にＴＮｅｔに送られるパケット送信機１２０に伝達される。次に、データ構造に含まれるチェイン・ポインタは、別のデータ構造にチェインで繋がれるこ

とが必要であるならば、アクセスされかつ作用を始動した適切なＢＴＥレジスタ３００、３０２に書込まれて、要求パケットを受信すべく次の素子のための次の構造に対するアドレスを供給する。

【０１２２】エンドーオブーリスト（ｅｌ）ビットは、セットされたときには、チェインの終りを示し、かつＢＴＥ処理を停止する。割込み終了（ｉｃ）ビットは、セットされたときには、インターフェイス装置２４ａに、前のＢＴＥ送信パケット（チェイン・ポインタによって示されたものではない）の終了を示すために割込みレジスタ２８０（図２１）にビットをセットする割込み（ＢＴＥＣｍｐ）をアサートさせる。割込みタイムーアウト（ｉｔ）ビットは、セットされたときには、インターフェイス装置２４ａに、アクセスの肯定応答がタイムーアウト（即ち、要求タイマ（図示省略）がタイムアウト信号を発行して、予期応答を適切な時間内に受け取らなかったことを示す）するか、またはＮＡＫ応答を導き出す（要求のターゲットが要求を処理できないということを示している）ならば、プロセッサ２０に対する割込み信号をアサートさせる。そして、チニック・サム（ｃｓ）ビットがセットされたならば、外部素子に書込まれるべきデータは、チェック・サム量を生ずるべくＢＴＥ８８（インターフェイス２４ａ；図９）のチェック・サム発生器（図示省略）を通過させる。生成されたチェック・サムは、メモリに書込まれ、それ自身のパケットに続いて位置決めさかつチェック・サムが形成されたデータを含んでいるメッセージ・パケットの宛先に送られうる。

【０１２３】纏めると、ＣＰＵ１２Ａのプロセッサ２０が外部装置にデータを送ることを望むときには、それらは、データ構造の部分３０４ａに識別子情報、バッファ部分３０４ｂにデータを含んでいる、データ構造３０４をメモリ２８に書込む。次に、プロセッサ２０は、データの優先度を決定し、かつデータ構造３０４（即ち、ヘッダ部分３０４ａ）を見出すことができるメモリ２８のアドレスを有するＢＴＥレジスタ３００，３０２を書込み、同時にＢＴＥレジスタ３００，３０２の要求終了ビット（ｒｃ１）をクリアして、ＢＴＥ動作を、ＢＴＥ状態マシン３０６の制御下で開始させる。Ｄｅｓｔ，ＴＹＰＥ，Ｌｅｎ，及び部分３０４ａからのアドレス情報は、メモリ２８からアクセスされかつ適切なパケット形に配置されるパケット送信機１２０に伝達される。データ構造３０４が、転送が書込み動作であることを指定するならば、局所バッファ・ポインタは、アクセスされかつデータ・バッファ部分３０４ｂを位置決めするために用いられる。次に、データは、アクセスされ、パケット送信機１２０に伝達され、ヘッダ及びアドレス情報と一緒にパケット化され、かつ送られる。データ構造３０４が読取り要求（例えば、プロセッサ２０が外部装置　－　Ｉ／Ｏ装置またはＣＰＵ１２のいずれかからのデータをシークする）を示すならば、Ｌｅｎ及びＬｏｃａｌ　Ｂｕｆｆｅｒ　Ｐｔｒ情報は、（外部素子から要求が行われたところに）読取り応答パケットが戻されたときに、メモリ２８への書込み要求を生成するために用いられる。データは、局所メモリ書込み動作が実行されるまでパケット受信機１００（図９）の入力パケット・バッファ１１０に保持される。

【０１２４】外部装置へのプロセッサー生成型読取り要求に対する応答は、ＡＶＴ表論理回路１４６によって処理されない。それよりも、プロセッサ２０がＢＴＥデータ構造をセットアップしたときに、トランザクション・シーケンス番号（ＴＳＮ）は、要求が割り当てられ、かつ上述したＨＡＣ型パケット（図６～図８）である、ＢＴＥ８８によって形成されかつ送られるメッセージ・パケットのヘッダ・フィールドに含まれる。また、プロセッサ２０は、データが、受信したときに、配置されるべきところで、ＢＴＥデータ構造においてメモリ・アドレスを含む。ＢＴＥ論理回路８８が進行してパケットを送るときには、バッファ位置のメモリ・アドレスは、レジスタ・ファイル（図示省略）に書込まれる、要求トランザクション論理回路１００（図９）であり、レジスタ・ファイルへのポインタとしてＴＳＮを用いている。応答（ＨＤＣメッセージ・パケットの形である　－　図７）がＣＰＵ１２によって受信されたときに、要求トランザクション論理回路１００は、入力メッセージ・パケットに含まれるデータがメモリ２８において配置されるバッファの対応メモリ・アドレスに対するレジスタ・ファイル（図示省略）へのポインタとしてパケットのヘッダからのトランザクション・シーケンス番号（ＴＳＮ）を用いる。

【０１２５】ＢＴＥレジスタ３００，３０２の優先順位(prioritization)を理解擦るために、ＣＰＵ１２Ａから外部装置へのデータの前述の転送は、情報の大きなブロックのものであると想定する。従って、多数のデータ構造は、プロセッサ２０によってメモリ２８にセットアップされ、（最後を除き）それぞれは、追加データ構造へのチェイン・ポインタ、送られるべきデータを（データ構造３０４のデータ・バッファ部分３０４ｂに）備える総計を含んでいる。優先順位の高い要求は、プロセッサ２０によってなされるのが望ましいということをここで想定する。そのような場合には、そのような優先順位の高い要求に対する関連データ構造３０４は、上述したのと同じ形で、メモリ２８に書込まれる。次に、優先順位の高いＢＴＥレジスタ３００は、データ構造を位置決めするために必要なＢＴＥアドレス、及びクリアされた要求終了表示ビット（ｒｃ０）で書込まれる。しかしならば、ＢＴＥレジスタ３００を書込むことによって示されたＢＴＥ要求は、すぐにはスタートしない。それは、ＢＴＥレジスタ３０２の内容によって始動されたＢＴＥ動作がパケット間で休止するまで待つ。更に、ＢＴＥレジスタ３０２の内容により信号が送られたＢＴＥ動作は、

ＢＴＥレジスタ３００の内容によって示されたＢＴＥ動作のために中断される。そのＢＴＥ動作は、終了するまで進行し、そのときに、ＢＴＥレジスタ３０２の内容によって信号が送られたＢＴＥ動作は、再開され、かつＢＴＥレジスタ３００が別のＢＴＥ動作記述子で再び書込まれない限り終了される。

【０１２６】－ メモリ・コントローラ：図４にちょっと戻ると、インターフェイス装置２４ａ，２４ｂは、一対のメモリ・コントローラ（ＭＣ）２６ａ，２６ｂを介してメモリ２８をアクセスする。Ｍｃｓは、インターフェイス装置２４とメモリ２８の間にフェイルーファースト・インターフェイスを供給する。Ｍｃｓ２６は、（ダイナミック・ランダム・アクセス・メモリ（ＤＲＡＭ）論理回路で実施される）メモリ・アレー２８をアクセスするために必要な制御論理回路を供給する。Ｍｃｓは、インターフェイス装置２４からメモリ要求を受信し、かつ読取り及び書込みを実行すると共にリフレッシュ信号を２８でメモリ・アレーを実施するＤＲＡＭｓに供給する。二つのＭｃｓ２６ａ，２６ｂは、メモリ・アレー２８とインターフェイス装置２４ａ，２４ｂとの間に７２ビット・データ経路を供給すべく並列に走り、ＳＢＣ－ＤＢＤ－ＳｂＤ ＥＣＣスキームを用いており、ここでｂ＝４であり、合計１００ビット（６４データ・ビット＋２８アドレス・ビット＋８チェック・ビット）のうち、７２ビット（６４データ及び８チェック・ビット）だけが実際にメモリ２８に書込まれる。

【０１２７】図２４を参照すると、メモリ２８から１４４ビットのデータをフェッチすべく並列に動作する二つのＭｃｓ２６ａ，２６ｂが示されている。一つのＭＣ（例えば、２６ａ）は、ＭＣとメモリ２８の間に７２ビット経路３３０ａを形成すべく８チェック・ビットと一緒に連続偶数アドレスで二つの３２ビット語を同時にアクセスすべく接続される。他のＭＣ（例えば、２６ｂ）は、第２の７２ビット経路３３０ｂを形成すべく二つの３２ビット奇数語を別の８チェック・ビットと一緒に同様にアクセスすべく接続される。この構成は、二つのＭｃｓ２６ａ，２６ｂを一緒に作動させ、かつ最小待ち時間でインターフェイス装置２４に６４ビット語を同時に供給させ、半分（Ｄ０）は、ＭＣ２６ａから生じ、他の半分（Ｄ１）は、他のＭＣ２６ｂから生ずる。インターフェイス装置２４は、ＥＣＣチェック・ビットを生成しかつ検査する。用いられるＥＣＣスキームは、（単一ビット）データ誤りを検出しかつ修正するだけでなく、全ての二重ビット誤り及び単一ＤＲＡＭからの４ビットまでの誤りも検出する。フェイルーファースト設計は、インターフェイス２４とＭＣバス２５の間、並びに内部レジスタ間のアドレス転送上のパリティを検査する。

【０１２８】インターフェイス装置２４の視点から、メモリ２８は、二つの命令でアクセスされる：“Ｎ二重語を読取る”及び“Ｎ二重語を書込む”。これら指令の両方は、第１の３６ビット転送上のアドレス及び制御と、第２の３２ビット転送上のバイト・カウントとを有するＭｃｓ２６に入力する。書込みで、Ｍｃｓ２６は、指令を、二重語書込み、または二重語書込みのブロックのいずれかに分解する。読取りで、要求データは、単一二重語読取りまたはブロック読取りフォーマットのいずれかに戻される。“データ有効”と呼ばれる信号は、読取りデータが戻されるかまたは戻されない２サイクル先の時間をインターフェイス装置２４に告げる。上記したように、保守プロセッサ（ＭＰ１８；図１）は、ＣＰＵｓ１２へのアクセスの二つの手段を有する。一つは、ＴＮｅｔ構造を用いることによって、パケット化された情報を送る（または受信する）ために、ルータ１４を含んでいる。より限定されたにもかかわらず、別のものは、システム１０の種々の素子（例えば、ルータ１４、ＣＰＵｓ１２、Ｉ／Ｏパケット・インターフェイス１６）に組込まれたＯｎ Ｌｉｎｅ Ａｃｃｅｓｓ Ｐｏｒｔ（オン・ライン・アクセス・ポート）（ＯＬＡＰ）を通してである。アクセスのこの後者の形は、メモリ・コントローラ２６のそれぞれを通してＭＰ１８に対する読取り及び書込みアクセスの両方を供給しているＯＬＡＰ直列ポート２８５を示す図２５に示されている。（図２５に示したのは、メモリ・コントローラ２６ａへのＯＬＡＰアクセスである；メモリ・コントローラ２６ｂは、実質的に同一デザインのものである。）ブート・タイムでは、ＭＰ１８は、プロセッサ２０に、それら（プロセッサ２０）に動作を開始させ、メモリに一連の命令のイメージを組込ませる命令で、ＯＬＡＰ２８５に含まれるレジスタを書込み、例えばブート処理を終了する外部（記憶）装置から命令及びデータを転送すべくＩ／Ｏに行く。

【０１２９】また、ＯＬＡＰ２８５は、誤り表示をＭＰ１８に伝達するためにプロセッサ２０によって用いられる。例えば、インターフェイス装置２４の一つが、メモリ・コントローラ２６から受信したデータにおいてパリティ誤りを検出したならば、それは、動作を一時停止する誤り信号を発行すると共に、誤りをＭＰ１８に知らせるためにビット位置をＯＬＡＰ２８５にセットさせる。メモリ・コントローラ２６によって実行される誤り検査（例えば、パリティは、レジスタ読取り動作上の不良を検出する）は、動作を同様に一時停止しかつ 誤りが発生したことをＯＬＡＰ２８５を介してＭＰ１８に知らせる。システムのＭＰ１８及び種々のＯＬＡＰｓ（例えば、ＭＣ２６ａのＯＬＡＰ２８５）は、ＩＥＥＥ標準１１４９．１に準じて構成される直列バス２８７を通して通信する。メモリ・コントローラのアーキテクチャは、Ｍｃｓ２６を実施することにおいて用いる種々の状態マシンを監視することによる誤り検査の特定の形を除き、一般に通常の設計のものである。図２６が示すように、ＭＣ２６ａの機能（ＭＣ２６ｂについても同様）は、それぞれが複写されかつ比較される、３つの主な状態マシ

ンによって制御される。マスタ状態マシン・ペア３９０は、メモリ２８にデータを伝達するためにMCADバス２５からDRAMデータ・バスにデータ及び命令を得るような、MC２６ａ自身の機能を制御すべく動作する。次に、マスタ状態マシン・ペア３９０は、MC２６ａと対応インターフェイス装置２４ａとの間のバス２５上のデータ及びアドレス転送を処理するメモリ制御アドレス／データ（MCAD）状態マシン３９２にわたり制御を実施する。DRAMデータ・バス上のアドレシング及びデータ転送、並びに必要なリフレッシュ信号の生成及びシーケンシングは、DRAM状態マシン・ペア３９４によって制御される。状態マシン・ペア３９０，３９２，及び３９４によって入力されたディジタル状態は、比較回路３９５によって互いに比較される。比較ミス(mis-compare)は、CPU１２の動作を一時停止すべく比較ミスを検出している比較回路３９５からのERROR信号のアサーションを結果として生ずる。

【０１３０】パケット・ルーティング：処理システム１０の種々の素子間（例えば、CPUs１２A，１２Bと I／Oパケット・インターフェイス１６に結合された装置）で伝達されるメッセージ・パケットは、パケットに含まれる情報（即ち、情報の他のものは、ソース・フィールドとしても、用いることができるが、ヘッダの宛先フィールド、図５（ｂ））により、ルータ１４によって“送られる(routed)”。しかしながら、ルータ１４の構成及び設計を説明する前に、CPUs１２とルータ１４との間、またはルータ１４とI／Oパケット・インターフェイス１６との間のTNetリンクL上でメッセージを伝達するために用いるプロトコルをまず理解することは、有利である。まず、各TNetリンクLは、受信及び送信能力（機能）の両方を有するポートを介して処理システム１０の素子（例えば、ルータ１４A）に接続する。素子の各送信ポートは、記号毎の、メッセージ・パケットの同期送信に用いられる送信クロック（T__Clk）信号を供給しなければならない。記号は、送信の受信端のクロック同期FIFOが同期を維持するようにT__Clkの各及び全てのクロック・サイクル（即ち、各クロック周期）で送信される。

【０１３１】クロック同期は、処理システム１０が動作されるモードに依存する。CPUs１２A及び１２Bが、例えば、互いに独立（個別）に動作するシンプレックス・モードで動作しているならば、ルータ１４とCPUs１２との間のクロッキングは、“近周波数”である；即ち、CPUs１２及びCPUsに直接接続するルータ１４によって用いられるクロックは、互いに関してドリフトしうる。逆に、処理システム１０がデュプレックス・モードで動作する（例えば、CPUsが同期された、ロックーステップ動作で動作する）ときには、それらが接続するルータ１４とCPUs１２との間のクロッキングは、“周波数封じ込み（位相封じ込みである必要はない）である。処理システム１０の種々の素子間のデータ・パケットのフローは、いつでも、パケット内でさえも、現れうる、指令記号によって制御される。（表１を参照して）上記で考慮したように、指令記号は、全て０である上位ビットによって識別される。それらの指令記号は、次の通りである。

【０１３２】IDLE：　IDLE指令記号は、送るべき他の指令記号またはデータ・パケットが存在しないときにクロック毎に送信される。IDLE指令記号は、TNetリンク上でパケットまたは指令記号間の空白詰め物(space filler)として作用する。

BUSY：　BUSY指令記号は、受信装置がデータ記号を受容することがすぐにできなくなるときに送られる。

FILL：　FILL指令記号は、それが記号を送っている受信素子が使用中であるということを送信素子が知る（例えば、BUSY指令記号の受信により）ときに送信素子によってメッセージ・パケットに注入される。

HALT：　この指令記号は、CPU１２またはMP１８によって始動（起動）され、かつ全てのCPUs１２及びある一定のI／O装置によるソフトウェア作用を必要とする事象を伝達すべくルータ１４により処理システム１０の全ての素子に発布される（広められる）。HALT指令記号は、I／Oアクティビティを始動（起動）することを停止することが必要であることをシステム１０の全てのCPUs１２に素早く通知する機構を供給する。

【０１３３】OTHER　LINK　BAD（OLB）：　CPU１２に接続されかつデュプレックス・モードで動作しているルータ１４がCPUs１２の一つから受信する指令記号またはパケットに誤りを検出し、かつCPUs１２の他のものから受信する指令記号またはパケットに誤りを検出しないときに、ルータ１４は、よい（良好な）パケットまたは指令記号を送付したCPU１２にOLB指令記号を送る。また、この指令記号は、デュプレックス・モードにおいてのみ、CRC誤り、指令記号誤り、及びプロトコル違反誤りに応じても送られる。OLB及びTLB（以下に説明する）指令記号は、デュプレックスされたCPUs１２へ同時に送られる；即ち、TLB指令記号は、そこから誤ったパケットまたは記号が受信され、または誤りに気付き(error noted)、かつそこから実質的に同時にOLB記号がデュプレックスされたペアの他のCPU１２に送られる、ようなCPU１２に送られる。

READY：　この指令記号は、先に使用中の素子がいま追加データを受容できるときに送られる。

SKIP：　この指令記号は、随意にスキップされうるクロック・サイクルを示す。この指令記号は、（１）各記号を転送し、かつそれを各受信クロック同期FIFOに装填し、かつ（２）FIFOからの記号を検索する、

二つのクロック信号間で同期を維持することへの補助として近周波数動作に関連して用いられる。

【０１３４】ＳＬＥＥＰ：　この指令記号は、ＲＥＡＤＹ指令記号（以下に説明する）が受信されるまで追加のパケット（もしあれば、現在送信されているものの後）が特定のリンクＬにわたり送られえないことを示すべく処理システム１０のあらゆる素子によって送られる。

ＳＯＦＴ　ＲＥＳＥＴ（ＳＲＳＴ）：　ＳＲＳＴ指令記号は、ＣＰＵｓ１２とルータ１４Ａ，１４Ｂとの間の記号転送を同期し、続いて、デュプレックス動作に対して同一の状態にＣＰＵｓ１２を配置するために用いられる処理（以下に説明する、“同期”及び“再統一(reintegration)”）の間中にトリガとして用いられる。

ＳＹＮＣ：　ＳＹＮＣ指令記号は、デュプレックス・モードに入る前に、または以下に完全に説明するように、デュプレックス・モードにおいて同期を要求するときに、ＣＰＵｓ１２とルータ１４Ａ，１４Ｂとのの間に周波数封じ込み同期を確立すべく処理システム１０（即ち、サブプロセッサシステム１０Ａ／１０Ｂ）のＣＰＵ１２へルータ１４によって送られる。ＳＹＮＣ指令記号は、例えば、Ｓｙｎｃｈｒｏｎｉｚａｔｉｏｎ及びＲｅｉｎｔｅｇｒａｔｉｏｎのセクションで以下に更に説明するように、システム・オペレーティング・モードを切り替える（即ち、シンプレックスからデュプレックスまたはデュプレックスからシンプレックス）ためにＳＲＳＴ指令記号に関連して用いられる。

【０１３５】ＴＨＩＳ　ＬＩＮＫ　ＢＡＤ（ＴＬＢ）：　ＴＮｅｔリンクＬから記号を受信しているシステム素子（例えば、ルータ、ＣＰＵ、またはＩ／Ｏ装置）が指令記号またはパケットを受信しているときに誤りに気付くときには、それは、ファシリティ・パケットまたは記号を送付したシステム素子にＴＬＢ指令記号を送り返す。それは、ＣＲＣ誤り、指令記号誤り、またはプロトコル違反誤りに応じて通常送られる。

Ｉ　ＯＷＮ　ＹＯＵ（ＩＯＹ）：　ＩＯＹ指令記号は、送信ＣＰＵからのデータを選択することをルータ１４に強要するためにルータ１４へＣＰＵ１２によってのみ（かつデュプレックス・モードで動作しているときにのみ）送られて、送信ＣＰＵ１２に、実質的に、所有権を与える；非送信ＣＰＵからの更なるデータ送信は、無視される。ＩＯＹ指令記号の実際のビット構造は、Ｏｔｈｅｒ　Ｌｉｎｋ　Ｂａｄ（ＯＬＢ）指令記号に用いられるものと同じである　－　記号のソースがどれかを決定する。ＩＯＹ／ＯＬＢ記号がＣＰＵ１２によって送られたならば、それは、ＩＯＹ記号として解釈される；ＩＯＹ／ＯＬＢ記号がルータによって送られたならば、それは、ＯＬＢとして解釈される。換言すると、ＣＰＵｓ１２とルータ１４Ａ，１４Ｂとの間で、ＣＰＵｓだけがＩＯＹ指令記号を送りかつルータだけがＯＬＢ指令記号を送る。

【０１３６】ＤＩＶＥＲＧＥ（ＤＶＲＧ）：　ＤＶＲＧ記号は、デュプレックス動作におけるときに、ＣＰＵｓから受信したデータ・ストリームの発散が検出されたことをデュプレックスされたＣＰＵｓに知らせるために、ルータによって送られる；即ち、ルータは、クロック同期ＦＩＦＯｓから引かれたときに互いに比較される記号の同一ペアを二つのＣＰＵｓ１２から受信している．．ＤＶＲＧ指令記号は、比較ミスに気付いたことをＣＰＵ１２に知らせる。ＣＰＵｓによって受信されたときに、発散検出処理が入力され、それによりどのＣＰＵが故障しているかまたは誤っているか、かつそのＣＰＵの更なる動作を終了することの決定がＣＰＵｓによってなされる。

ＴＨＩＳ　ＰＡＣＫＥＴ　ＧＯＯＤ（ＴＰＧ）：　パケットの送信者がパケットのＣＲＣが良好であると決定したことを示している、メッセージ・パケットに続く指令記号。より詳細は、以下の“Ｐａｃｋｅｔ　Ｓｔａｔｕｓ”を参照のこと。

ＴＨＩＳ　ＰＡＣＫＥＴ　ＢＡＤ（ＴＰＢ）：　ＴＰＢ指令記号は、受信素子が受信メッセージ・パケットのＣＲＣが正しくないということを決定したときにＴＰＧ指令記号を置換する。

【０１３７】－　Ｆｌｏｗ　Ｃｏｎｔｒｏｌ：ルータ１４は、記憶容量が限られており、従って、メッセージ・パケットをルーティングするときには、“蓄積交換(store and forward)”方法の型を用いない；それよりも、それらは、“虫の穴(worm-hole)”ルーティングとして知られるものを実施する：メッセージ・パケットのヘッドは、そのテールが受信される前にルータを通過してそれから出る。これは、上述したＢＵＳＹ／ＦＩＬＬ／ＲＥＡＤＹ指令記号を主に用いて、上述した指令記号が処理システム１０の種々の素子間（例えば、ＣＰＵｓ１２、ルータ１４、等）のメッセージ・フローを制御すべく動作する一つの理由である。このフロー制御は、“バックプレッシャ”と称される。特定のシステム素子は、その受信キュー（例えば、エラスティック（伸縮性）・バッファ５０６　－　図２７）がほとんどいっぱい（充満）であることを決定するときはいつでも、それは、その上でそれが入力メッセージ・パケットを受信しかつ、更なる送信を防ぐことを送信素子に告げるべく関連送信ポートからのＢＵＳＹ指令記号を、ＴＮｅｔリンクＬの他端の送信素子に、送信する、ＴＮｅｔリンクＬの双方向能力（機能）を利用する。ＢＵＳＹ指令記号の使用は、“バックプレッシャ”をアサートすると、ここで称される。ＣＰＵｓ１２またはＩ／Ｏパケット・インターフェイス１６は、その送信ポートの一つがバックプレッシャされるので“エンド・ノード(end node)”（例えば、ＣＰＵ１２またはＩ／Ｏ装置１７　－　図１～図３）がバックプレッシャをアサートしえないが、そのような内部資源が特定のＴＮｅｔポートにアサートされた

バックプレッシャに関係なく利用可能になるときにのみ、内部資源が利用可能になるのを待っている間にそのようなバックプレッシャをアサートしうる。この要求事項（必要事項）の監察することの失敗は、送信ポートが送信できないので、そしてまた関連受信機がバックプレッシャをアサートしているので、受信ポートが受信できないところのバックプレッシャ・デッドロック(backpressure deadlocks)を結果として生じうる。それゆえに、ルータ１４だけがバックプレッシャを伝播できる；エンド・ノード（ＣＰＵｓ１２，Ｉ／Ｏパケット・インターフェイス１６）は、受信バックプレッシャを送信バックプレッシャに変換することを許可されていない。

【０１３８】ルータ１４は、そのポートに到着している更なるデータ記号がバッファされるか前進されることができないときはいつでも、その受信ポートのいずれか一つにバックプレッシャをアサートしうる。不適当にアドレスされたパケットは、ルータ１４によって廃棄される。処理システム１０のシステム素子がその上でそれがメッセージ・パケットを送信しているＴＮｅｔリンクＬ上のＢＵＳＹ指令記号を受信したときには、素子は、パケットを送ることを停止し、かつＲＥＡＤＹ指令記号が送信クロックＴ＿Ｃｌｋの各クロック・サイクルで受信されるまで、その代わりにＦＩＬＬ指令記号を送ることを始める。ＦＩＬＬ指令記号は、送られ続ける。また、関連送信ポートがパケットを送信しない間にＢＵＳＹ指令記号がＴＮｅｔリンクＬ上で受信されるならば、ＢＵＳＹ記号を受信している素子は、それがそのリンク上でＲＥＡＤＹ記号を続いて受信するまで新しいパケット送信を始動することを慎む。送信ポートは、他の指令記号（ＲＥＡＤＹ，ＢＵＳＹ等）を送信する能力（機能）をさもなくば保持する。処理システム１０の素子のＴＮｅｔポートがＲＥＡＤＹ指令記号を検出するときはいつでも、それは、関連送信ポートでのＦＩＬＬ指令記号の送信を終了し、かつ先の受信ＢＵＳＹ指令記号によって停止されたパケットを送くることを再開するか、またはそれは、ＩＤＬＥ指令記号を注入することを終了しかつ保留しているパケットを送ることを始動するか、またはパケットが利用可能であるまでＩＤＬＥ指令記号を送り続ける。

【０１３９】しかしながら、ＢＵＳＹ／ＲＥＡＤＹフロー制御は、他の指令記号の送信に適用されないということが理解されるべきである。上述したように、送信クロック、Ｔ＿Ｃｌｋ、の全てのサイクルは、指令またはデータ記号の送信を伴うことを思い出す。それゆえに、全てのＴＮｅｔインターフェイスは、ＴＮｅｔインターフェイスが受信する、関連送信クロック、Ｔ＿Ｃｌｋ、のクロック・サイクルで新しい指令またはデータ記号を受容する準備ができていなければならない。分かるように、送信された記号を受信するためにＴＮｅｔリンクＬに接続する処理システム１０の全ての素子（例えば、ルータ１４、ＣＰＵｓ１２）は、クロック同期（ＣＳ）ＦＩＦＯを介してそれらの記号を受信する。例えば、上述したように、ＣＰＵｓ１２のインターフェイス装置２４は、全てのＣＳ　ＦＩＦＯｓ１０２ｘ，１０２ｙ（図１０に示される）を含む。各ＣＳ　ＦＩＦＯ１０２は、対応ＴＮｅｔリンクＬから指令またはデータ記号を受信すべく接続される。ＣＳＦＩＦＯは、スピード・マッチング（速度整合）を許容すべく十分な深さを供給しなければならないし、かつエラスティックＦＩＦＯｓは、メッセージ・パケットの受信の間中のＢＵＳＹ指令記号の送信と、ＦＩＬＬまたはＩＤＬＥ指令記号のためによる入力メッセージ・パケットの中断との間で発生しうる遅延を処理するのに十分な深さを供給しなければならない。また、ルータ１４のエラスティックＦＩＦＯｓ５０６（図２７）は、送信経路におけるＢＵＳＹ及びＲＥＡＤＹ指令記号の注入を許容すべく十分な深さを供給すべきである。例えば、図１～図３を参照すると、ＣＰＵ１２Ａが、ルータ１４Ａのポート３を介して　－　Ｉ／Ｏパケット・インターフェイス１６Ａの一つによる受信に対してメッセージ・パケットを送信していることを想定する。同時に、また、ＣＰＵ１２Ａによって送られるメッセージ・パケットを受信しているその同じＩ／Ｏパケット・インターフェイス１６Ａは、ルータ１４Ａのポート３に同じ（双方向）ＴＮｅｔリンクＬ上のメッセージ・パケットを送っている。ルータ１４ＡがＩ／Ｏパケット・インターフェイス１６Ａによって送られるメッセージ・パケットの宛先からのホールドアップ（バックプレッシャ）を経験することを更に想定する。ある時間後、エラスティックＦＩＦＯ５１８（図２７）は、メッセージ・パケットの送信を一時的に停止すべくＩ／Ｏパケット・インターフェイスにリクエストすることをルータ１４Ａに要求する点まで充たす。従って、ルータ１４Ａは、ポート３（Ｉ／Ｏパケット・インターフェイス１６Ａからのメッセージ・トラフィックを受信しているのと同じポート）からＢＵＳＹ記号を送信する。そのＢＵＳＹ記号は、ＣＰＵ１２Ａからルータ１４Ａを通って送られるメッセージ・パケットの記号ストリームに挿入される。入力メッセージ・パケットのストリームへのＢＵＳＹ記号の挿入は、入力パケットの一つの余分な記号を記憶することをルータ１４Ａに要求する。ＢＵＳＹ記号が送られた後、ルータ１４Ａは、それがＩ／Ｏパケット・インターフェイス１６Ａからのメッセージ・パケットの切断された送信の受信を再び始められるような時までＣＰＵ１２Ａからの入力メッセージ・パケットの送信を再開することができる。Ｉ／Ｏパケット・インターフェイス１６Ａに対して割込まれたメッセージ・パケットの再送信を始めるために、ルータ１４Ａは、ポート３から送られる記号ストリームにＲＥＡＤＹ信号を挿入し、ＣＰＵ１２Ａからのメッセージ・パケットの別の記号を記憶することをルータに再び要求する。

【０１４０】ＢＵＳＹ／ＲＥＡＤＹ指令記号のこのペアは、Ｉ／Ｏパケット・インターフェイス１６とＣＰＵ１２との間の経路に各ルータ１４及びＣＰＵ１２によって挿入することができる。Ｉ／Ｏパケット・インターフェイス１６に直接接続されたルータ１４は、２ｎ指令記号（ｎ＝経路におけるルータの数＋１）を単一パケットに注入できる。これは、一方向にアサートされたバックプレッシャが、反対方向にアサートされるためにバックプレッシャを必要としないことを確実にするために２ｎバイトのＦＩＦＯが最低レベル・ルータ１４（即ち、Ｉ／Ｏパケット・インターフェイス１６に最も近いルータ）に要求されるということを意味する。例えば、Ｉ／Ｏパケット・インターフェイス１６がルータ１４にパケットＡを送信し、同時にその同じルータからパケットＢを受信し、パケットＡを受信しているそのルータがバックプレッシャによりそれを先に進めることができないものと想定する。そのルータは、パケットＡを送ることを停止すべくＩ／Ｏパケット・インターフェイス１６に知らせるためにＢＵＳＹ信号をパケットＢに注入しなければならない。パケットＢに注入されたＢＵＳＹ指令記号は、一つだけＦＩＦＯの深さを増加する一つのデータ記号を変位する。ＲＥＡＤＹを注入することによるバックプレッシャの後続の除去は、パケットＢにおける別のデータ・バイトを変位する。パケットＡが次のルータに進むと、処理は、繰り返される。ルータ１４がＦＩＦＯが処理できるよりも多くのデータ・バイトを変位したならば、それは、パケットＢのソースにバックプレッシャをアサートしなければならない。

【０１４１】－　パケット状態：各送信されたパケットは、ＴＰＧまたはＴＰＢ指令記号がすぐ後に続き、関連パケットのインテグリティを報告する。パケットがそこで始まるシステム素子は、適切なＴＰＧまたはＴＰＢ指令記号を挿入する。ルータ１４は、付随しているＣＲＣを確認し、かつそれらがソース（例えば、Ｉ／Ｏパケット・インターフェイス１６またはＣＰＵ１２）から宛先（例えば、ＣＰＵ１２Ａまたは Ｉ／Ｏパケット・インターフェイス１６）にフローするときに全てのパケットに対して種々のプロトコル検査を実行する。問題のフローの経路における、ルータ１４が入力パケット上に誤りを検出し、かつパケットがＴＰＧ指令記号で終結する（パケットが良好であることを示している）ならば、ルータは、ＴＰＧ指令記号をＴＰＢ指令記号で置換する。ＴＰＢ記号へのＴＰＧ指令記号の変更をもたらすことができる誤りは、検査したときにＣＲＣデータによる受信データを確認することの失敗に加えて、使用したプロトコルによって許容されるものよりも大きい長さを有するパケットを含む。あらゆるパケット長を用いることができるが、ここでは、パケットは、状態（ＴＰＧ／ＴＰＢ）記号を含んで、１０２４記号に制限される。受信したパケットが、この制限よりも多くを有しているとして検出されたならば、受信ルータは、１０２４番目の記号におけるＴＰＢ指令記号でパケットを終了し、パケットの残りを廃棄する。パケット長のこの制限は、それを絶え間なくバブルさせ、かつＴＮｅｔネットワークを詰まらせるパケット送信素子において発生することからフォルトを排除する誤り検査技術である。

【０１４２】ＴＰＢ指令記号に続いてパケットを受信するルータ１４は、それ自身の結果にかかわらず、不変更のＴＰＢ指令記号を先に進める。

【０１４３】－　ＳＬＥＥＰ　Ｐｒｏｔｏｃｏｌ：ＳＬＥＥＰプロトコルは、以下に説明する、保守インターフェイス（オンーライン・アクセス・ポートーＯＬＡＰ）を介して保守プロセッサによって始動される。ＳＬＥＥＰプロトコルは、パケット境界で一つ以上のＴＮｅｔリンクＬを静止する(quiesce) ための機構を供給する。一つのシステム１０を再統一するためにモードを変える（例えば、デュプレックスからシンプレックス）ことが必要である。ルータ１４は、データ損失または汚染をもたらすことなくモードを変えるためにアイドル(idle)（処理にパケットがない）でなければならない。ＳＬＥＥＰ指令記号を受信したときに、処理システム１０の受信素子は、そのＴＮｅｔリンクＬ上の許可された指令記号だけを送信しなければならない関連送信ポート上の新しいパケットの送信の始動を抑制する。（例外は、再統一を取り扱うセクションで以下に説明する、自己アドレス型ＡｔｏｍｉｃＷｒｉｔｅメッセージ・パケットである。）ＳＬＥＥＰ指令記号が受信されるときに送信されるパケットは、終了するまで普通に送信される。しかしながら、ＳＬＥＥＰ指令記号が受信される受信ポートに関連した送信ポートは、許可された指令記号（例えば、ＢＵＳＹ，ＲＥＡＤＹ，ＩＤＬＥ，ＦＩＬＬ）を送信し続けるが、ＲＥＡＤＹ指令記号がその関連受信ポートで受信されるまで送信のための新しいパケットを始動しない。

【０１４４】－　ＨＡＬＴ　Ｐｒｏｔｏｃｏｌ：ＨＡＬＴ指令記号は、Ｉ／Ｏアクティビティ（即ち、ＣＰＵｓ１２とＩ／Ｏパケット・インターフェイス１６との間のメッセージ送信、または異なるＣＰＵｓ１２間のメッセージ送信）を終了ことが必要であることを処理システム１０の全てのＣＰＵｓ１２に素早く知らせる機構を供給する。各ルータ１４は、ＨＡＬＴ指令記号がＣＰＵ１２から受信されたときに、受信ルータ１４がその送信ポートのそれぞれからＨＡＬＴ指令記号を伝播し、かつそのシステム停止イネーブル・ビットをクリアするようにＯＬＡＰ２８５’（図２７）を通してＭＰ１８によってセットすることができるシステムＨＡＬＴイネーブル構成レジスタを有する。ルータ１４は、システム停止イネーブル・ビットがクリアされた状態であるときに受信する全てのＨＡＬＴ指令記号を無視する。このようにして、システム停止イネーブル・ビットは、停止機能のための

ソフトウェア・セット可能イネーブル、並びに一度第１のＨＡＬＴ指令記号がアサートされるとＨＡＬＴ指令記号の無限サイクリングを防ぐことの両方に機能する。

（インターフェイス装置２４）のそれらの受信ポートのいずれかでＨＡＬＴ指令記号を受信するＣＰＵｓは、システム停止割込みがイネーブルされる（即ち、マスク・レジスタ２８２の関連処理(associated disposition)が割込みをイネーブルする；図２１）ならば割込みレジスタ２８０に割込みをポストする。

【０１４５】ＣＰＵｓ１２は、ＨＡＬＴ処理をディスエーブ（禁止）するための機能（能力）を備えうる。それゆえに、例えば、インターフェイス装置２４の構成レジスタ７５は、所定の状態（例えば、ＺＥＲＯ）にセットされたときにＨＡＬＴ処理をディスエーブルするが、誤りとしてＨＡＬＴ記号の検出を知らせる、“停止イネーブル・レジスタ”を含むことができる。

【０１４６】ルータ・アーキテクチャ：ここで図２７を参照すると、ルータ１４Ａの略ブロック図が示されている。処理システム１０の他のルータ１４（例えば、ルータ１４Ｂ，１４’等）は、実質的の同一な構成のものであり、従って、ルータ１４Ａに関する説明は、他のルータ１４に同様に適用する。図２７に示すように、ルータ１４Ａは、それぞれがポート入力５０２（502_0，．．．，502_5）及び出力５０４（504_0，．．．，504_5）を含んでいる、６つのＴＮｅｔポート０，．．．，５を含む。各ポート出力５０４は、上述したそれから出ている１０の信号回線を有する：並列９ビット指令／データ記号を送信する９つの信号回線、及び関連送信クロック（Ｔ＿Ｃｌｋ）を運ぶ信号回線。同様に、ポート入力５０２のそれぞれは、データ、受信クロック（Ｒｃｖ　Ｃｌｋ）を含んでいる１０の並列信号を受信すべく接続する。更に示すように、各ポート入力５０２は、入力論理回路５０５及びそれをクロスバー・スイッチ５００に印加する前に入力メッセージ・パケットを受信しかつバッファするエラスティックＦＩＦＯ５０６を含む。クロスバー論理回路５００は、メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎ（宛先）ＩＤに含まれる情報によりポート出力５０４へポート入力５０２によって受信したメッセージ・パケットを送るべく動作する。クロスバー論理回路５００は、真のクロスバー・スイッチとして動作して、そのポート出力５０４がパケットを受信しているポート入力５０２に関連したとしても（例えば、ポート入力502_2及びポート出力504_2）、ポート入力５０２で受信したメッセージ・パケットをポート出力５０４に送らせる。クロスバー論理回路５００は、ポート入力５０２の対応するものからポート出力５０４に二つ以上のメッセージ・パケットを送るようにも構成されている。クロスバー論理回路５００は、その構成についての更なる説明が必要でないように通常の設計のものである。

【０１４７】シェーディングにより図において強調された、ルータ１４Ａのポートの二つ、４及び５は、他のものとはある程度異なって構成される；これら二つのポートは、一対のＣＰＵｓ１２に（ＴＮｅｔリンクＬｘ及びＬｙによって）直接接続するそれらのポートとして用いられることを意図している。これらのポート４，５に対するポート入力502_4，502_5は、処理システム１０がデュプレックス・モード動作に対してセットされるときに周波数封じ込み環境で動作すべく構成される。更に、デュプレックス・モードであるときに、入力ポート０－５のいずれか一つで受信され、かつルータが接続するＣＰＵｓ１２のいずれか一つに向けられたメッセージ・パケットは、クロスバー論理回路５００によって繰り返え（再現）され、かつそれらが実質的に同時に、記号毎に、同じ記号を接続するＣＰＵｓへ送信すべくロックーステップ・ファッションで動作する二つのポート出力504_4，504_5の両方に送られる。デュプレックス・モードで動作していない（即ち、シンプレックス・モード）ときには、ポート入力502_4，502_5、及び全ての他のポート入力は、近周波数モードで動作する。更に、また、ルータ４及び５に対する入力論理回路５０２は、二つのＣＰＵｓから受信した指令／データ記号の記号毎の比較を実行すべく、ＣＰＵｓ１２Ａ，１２Ｂがデュプレックス・モードであるときに、動作する比較回路を備えている。従って、図２８に示すように、ポート入力502_4，502_5は、ＣＰＵｓから指令／データ記号を受信し、クロック同期ＦＩＦＯｓ５１８（以下に更に説明する）を通してそれらを渡し、かつクロック同期ＦＩＦＯｓを出て行く各記号をゲート型比較回路５１７と比較する。デュプレックス動作が入力されたときに、制御論理回路５０９の構成レジスタ（図示省略）は、ＤＰＸ信号をアサートする状態にセットされる。ＤＰＸ信号は、ポート４及び５に対するルータ入力論理回路５０２の二つの同期ＦＩＦＯｓ５１８から出て行く記号の記号毎の比較をアクティベート（活性化）すべく制御論理回路５０９からゲート型比較回路５１７へ伝達される。もちろん、ＤＰＸビットが制御論理回路５０９にセットされなかったときには、比較がディスエーブルされる。

【０１４８】同一記号ストリームである、デュプレックスされたＣＰＵｓ１２からのメッセージ・トラフィックは、ポート入力502_4，502_5によって受信され、一つのポート入力によって受信されたストリームの各記号は、他のポート入力によって、実質的に同時に、受信されたものと同一である。デュプレックス・モードにおいて同期を維持するために、ＣＰＵｓ１２に送信するルータ１４Ａの二つのポート出力は、ロックーステップで動作しなければならない；即ち、ポート出力は、サイクル間に応じて(on a cycle-to-cycle basis) 同じ記号がＣＰＵｓ１２の両方に送られなければならないように動

作しなければならない。それゆえに、図４を参照すると、ルータ１４Ａのポート０－５（図２７）の一つで受信され、かつＣＰＵｓ１２に向けられた記号ストリームは、同一記号が実質的に同時にＣＰＵｓによって受信されるように、デュプレックス動作で、両方のＣＰＵｓ１２に進められ（送られ）なければならない。（ＣＰＵｓ１２は、デュプレックス・モードであるときには、ルータ１４によって複写され、かつ両方のＣＰＵｓに戻される、自己アドレス型メッセージ・パケットを送ることができる。）ＣＰＵｓ１２に直接結合される出力論理回路装置５０４_4，５０４_5は、両方ともに同期したファッションで（メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎフィールドがデュプレックスされたＣＰＵｓ１２の一つ、例えば、ＣＰＵ１２Ａだけを識別しても）クロスバー論理回路５００から記号を受信し、二つのＣＰＵｓ１２へ実質的に同期ファッションでそれらの記号を与える。もちろん、ＣＰＵｓ１２（より正確には、関連インターフェイス装置２４）は、図１１に示したものと実質的に同じ構成の同期ＦＩＦＯｓで送信された記号を受信して、それを伴って記号がＣＰＵｓ１２によって受信される多少の実時間位相差が存在していても、二つのＣＰＵｓ１２間で維持されるクロッキングは、同じ記号が同じ命令サイクルで両方のＣＰＵｓ１２によってＦＩＦＯ構造から引き出されることを確実にして、デュプレックス動作モードによって要求されるＣＰＵｓ１２の同期した、ロックーステップ動作を維持する。

【０１４９】ポート入力５０２のより詳細な図（図２９及び図３１）の説明に関して見られるように、ルーティング制御は、（オン・ライン・アクセス・ポート２８５’及び直列バッファ１９Ａを介して；図１参照）保守プロセッサ１８により制御論理回路５０９に含まれるレジスタに書込まれた構成データと共に、ポート入力５０２の論理回路によって主に行われる。ルータ１４Ａは、適切な動作を確実にするためにルータ14A を構成する種々のコンポーネントの検査を実行する自己検査論理回路５１１を更に含む。一般に、自己検査論理回路５１１は、内部パリティ検査、状態マシンの違法状態検出、及び複写された論理回路の出力の比較のような動作を実行する。実行される自己検出は、通常の特質のものである。ルータ１４Ａの同期動作は、クロック論理回路５１０によって生成される（局所）検査信号により実行される。ルータ１４の各出力ポート５０４は、ＴＮｅｔリンクＬ上の記号を伝達するために、上述した、フロー制御プロトコルの要求事項を実施すべく構成される。また、各ポート入力５０２の入力論理回路５０５は、受信したＳＫＩＰ指令記号を除去することによって　－　少なくとも近周波数環境で記号を送っているポートに対して　－　同期を維持することを補佐する。ＳＫＩＰ指令記号は、このコンテキストにおいて、実質的に、クロック・サイクルをスキップさせて低速な受信機に高速な受信機からデータを受容させる、位置－保持記号として用いられる。ＴＮｅｔリンクＬの端末における装置が異なるクロックで動作するので、近周波数環境で動作しているときに、一つのクロックが多少の量だけ他のものよりも速いということは、相対的に確実である。検査されないままであったならば、高速－送信素子から記号を受信している低速－受信素子は、低速－受信素子の入力クロック同期ＦＩＦＯに負荷を掛け過ぎる。即ち、高速クロックによりそこに置かれた低速クロックがクロック同期ＦＩＦＯから記号を引く(pull)ために用いられたならば、結果として検査同期ＦＩＦＯは、オーバーフローするであろう。

【０１５０】ここに採り入れられた好ましい技術は、同期化ＦＩＦＯから記号を引くために用いる局所クロックよりも周波数が多少高いＦＩＦＯに記号を押す(push)Ｔ＿ＣＬＫ信号によりルータ１４（またはＣＰＵ１２）のクロック同期ＦＩＦＯ（即ち、クロック同期ＦＩＦＯ５１８；図２９）のオーバーフローの可能性を回避、または少なくとも最小にすべく記号ストリームにＳＫＩＰ記号を周期的に挿入することである。（ＦＩＦＯへの）押し(push)動作をバイパスするためにＳＫＩＰ記号を用いることは、ＳＫＩＰ指令記号が受信される毎にＦＩＦＯの押しポインタを失速させる効果を有するので、クロック同期ＦＩＦＯに関する限り、ＳＫＩＰ記号を伴った送信クロックが失われていた（欠けていた）。それゆえに、ポート入力５０２のそれぞれにおける論理回路は、何もＦＩＦＯに押されないが、記号が引かれるように、近周波数クロッキング環境における同期に対してＳＫＩＰ指令記号を認識し、かつその受信をキー・オフ(key off)する。ＳＫＩＰ記号は、おおよそ５１２送信機クロック毎に挿入されるのが好ましい。５０Ｍｈｚ速度で記号がリンクＬに送信される（例えば、ＣＰＵ１２とルータ１４との間、またはルータ１４間、またはルータ１４とＩ／Ｏインターフェイス装置１６Ａとの間　－　図１～図３）と仮定すれば、これは、最悪の場合、２０００ｐｐｍの周波数差を許容する。

【０１５１】各ポート入力５０２のエラスティックＦＩＦＯｓ５０６は、通常設計のものであり、例えば、通過中にメッセージ・パケットの中にフロー制御及び指令記号を挿入することによってもたらされた、記号ストリームにおけるジッタを吸収しかつ平滑にすることを補助するために用いられる。たぶん、最も重要なことは、エラスティックＦＩＦＯｓ５０６は、出力ポートが使用中のときに入力メッセージ・トラフィックのバッファリングを許容する。システム１０の他の素子のように、ルータ１４Ａは、宛先装置へ受信したメッセージ・パケットを送るときに“バックプレッシャ”を経験しうるし、かつ宛先装置は、更なる記号を受信できないことを瞬間的に知らせる（例えば、ＢＵＳＹ指令記号）。バックプレッシャの適切な実施は、エラスティックＦＩＦＯｓ５０６

が、先の装置（例えば、ルータにメッセージ・パケットを供給する装置）が（受信されかつクロック同期ＦＩＦＯｓに押されるが、エラスティックＦＩＦＯｓに渡されない）ＦＩＬＬまたはＩＤＬＥ記号を供給することによりＢＵＳＹ記号に応答できるまで宛先装置が受信を停止した後で入力記号を受信しかつ保持するのに十分に大きな深さ有する（即ち、十分な数の記憶位置を有する）ことを必要とする。要約すると、各エラスティックＦＩＦＯ５０６は、送信装置が送くることを一時的に停止できるまで記号を記憶し続けるために十分な空間をもたなければならない。

【０１５２】記号ストリームにおけるジッタを低減することを補助するために、エラスティックＦＩＦＯｓ５０６は、高及び低“水位標(water marks)”で動作する。エラスティックＦＩＦＯ５０６が充満し始め、かつ高水位標に達したならば、バックプレッシャ記号（例えば、ＢＵＳＹ）は、記号ストリームを受信している受信ポートに対応している送信ポートから送信される。例えば、記号ストリームがルータ・ポート入力５０２_3によって受信されており、かつエラスティックＦＩＦＯ５０６_3を制御するために用いられるＦＩＦＯ制御論理回路５４６がＦＩＦＯが充満している（即ち、高水位標を通過した）ことを示すならば、入力ポート５０２_3は、ＢＵＳＹ記号を送信させるべく対応出力ポート５０４_3に知らせる。ＢＵＳＹ状態は、ポート出力５０４_3がＲＥＡＤＹ記号を送るべく知らされて、記号ストリームのフローの再開を要求するときに、ＦＩＦＯ制御論理回路５４６（図２９）によって決定されたように、エラスティックＦＩＦＯ５０６_3の深さが低水位標以下であるまでルータ１４（及びパケットを送っていたＴＮｅｔリンクＬの他端における装置）によって維持される。ＴＮｅｔリンクＬの他端において、メッセージ・パケットを送っていた装置は、関連出力指令リンク上に送信されたＦＩＬＬ指令記号で入力リンクでのＢＵＳＹ指令記号の受信に応答する。送信装置は、ＢＵＳＹ指令記号を送った装置がＲＥＡＤＹ記号を送るまで、メッセージ・パケットの更なる送信を手控えて（見合わせて）、ＦＩＬＬ記号を送り続ける。メッセージ・パケットの送信は、終了まで、またはバックプレッシャが受信機によって再びアサートされるまで、再開する。

【０１５３】エラスティックＦＩＦＯｓ５０６がこの“バックプレッシャ”ジッタを処理するために十分に大きくなければならないばかりでなく、それは、制御記号が他の方向におけるＴＮｅｔリンクＬの制御に対して記号ストリームに挿入されている間にＦＩＦＯに累積するデータ記号を記憶することもできなければならないということに注目すべきである。ＢＵＳＹ／ＲＥＡＤＹ組合せは、ポート出力５０４から２サイクルを奪ってそのポート出力５０４を供給しているエラスティックＦＩＦＯ５０６を２文字(two characters)でいっぱいにする。ジッタを最小に保つために、エラスティックＦＩＦＯｓ５０６のサイジング（及び高及び低水位標の配置）は、バックプレッシャがアサートされる前にストリームの中に挿入されることを少なくとも二つの文字、好ましくはそれ以上に対して許容しなければならない。ここに記載されたシステムの環境内で、エラスティックＦＩＦＯｓ５０６は、９６記号を一時的に記憶することが可能である。ルータ１４Ａは、バックプレッシャが要求される前に挿入されることを所定数の記号に対して許容する（バックプレッシャは、所定数が受信されかつ一時的に記憶された後で次の記号に発行される）。エラスティックＦＩＦＯｓ５０６の９６記号深さは、所定数の記号の標準蓄積(buildup)、及びポート入力５０２がデータを受容することをやめる（バックプレッシャをアサートする）か、またはオーバーフローによるデータの損失の醜行(ignominy)に見舞われなければならない前にバックプレッシャ遅延の１２サイクルを許容する。

【０１５４】ポート入力５０２のそれぞれは、一つの説明が全てに適用されるように実質的に同一に構成される。従って、図２９に示すように、ポート０に対するポート入力５０２_0の詳細ブロック図が示される。ポート入力５０２_0は、それが付随する送信クロック（Ｔ＿Ｃｌｋ）によって一時的に記憶される入力レジスタ５１６で各９ビット・データ／指令記号を受信する。次に、受信した記号は、またＴ＿Ｃｌｋによって、入力レジスタ５１６から伝達されかつクロック同期化ＦＩＦＯ５１８に印加される。クロック同期ＦＩＦＯ５１８は、ＣＰＵｓ１２のインターフェイス装置２４に用いられる、図８Ａ及び図８Ｂに示したものと論理的に同じである。ここで、図２９が示すように、クロック同期ＦＩＦＯ５１８は、入力レジスタ５１６の出力を、並列に、受信する複数のレジスタ５２０を含んでいる。レジスタ５２０のそれぞれと関連するのは、図３０に詳細に示され、かつ以下に説明される、２段階妥当性（Ｖ）ビット・シンクロナイザ５２２である。レジスタ２０のそれぞれの内容は、各関連２段階妥当性ビット・シンクロナイザ５２２の１ビット内容と一緒に、マルチプレクサ５２４に印加され、かつ選択されたレジスタ／シンクロナイザがＦＩＦＯから引かれ、かつ一対のレジスタ５２６によりエラスティックＦＩＦＯ５０６に結合される。入力レジスタ５１６の内容を受信するレジスタ５２０の選択は、押しポインタ論理回路装置５３０によって供給されたＰｕｓｈ　Ｓｅｌｅｃｔ信号の状態によって決定される；そして、レジスタ５２６に、ＭＵＸ５２４を介して、その内容を供給するレジスタ５２０の選択は、引きポインタ論理回路５３２によって供給されたＰｕｌｌ　Ｓｅｌｅｃｔ信号の状態によって決定される。押し及び引きポインタ論理回路５３０，５３２は、ｓｙｎｃＦＩＦＯ制御論理回路５３４の制御下である。ｓｙｎｃＦＩＦＯ制御論理回路５３４は、押しポインタ論理回路５３０（並びに

入力レジスタ５１６）を動作しかつ押しポインタ論理回路５３０によって選択されたレジスタ５２０を装填すべく入力Ｔ__Ｃｌｋを受信する。同様に、同期ＦＩＦＯ制御論理回路５３４は、引きポインタ論理回路５３２を制御すべくルータ（Ｒｃｖ　Ｃｌｋ）に対して局所的であるクロック信号を受信する。

【０１５５】ちょっと脇道にそれて、図３０を参照すると、イネーブル機能を有するD型フリップフロップ５４１、遅延素子５４１ａ、ＯＲゲート５４１ｂ、（以下に示す真理値表に示される機能を供給するためのセット／リセット／イネーブル能力を有する）D型フリップフロップ５４２、及びD型フリップフロップ５４３を含んでいる妥当性ビット・シンクロナイザ５２２が詳細に示されている。D型フリップフロップ５４１は、そのデータ（D）入力でＳＫＩＰ検査論理回路５４０の出力を受信すべく結合される。フリップフロップ５４１のＥｎａｂｌｅ（イネーブル）入力は、押しポインタ５３０、Ｐｕｓｈ　Ｓｅｌｅｃｔ（選択）、によって供給される復号を受信し、かつフリップフロップ５４１のクロック（Ｃｌｋ）は、入力記号を付随している入力送信クロック（Ｔ__Ｃｌｋ）を受信する。フリップフロップ５４１の出力（Ｑ）は、ＯＲゲート５４１ｂの一つの入力に印加され、かつ遅延素子５４１ａを通して他の入力にも印加される。フリップフロップ５４１の出力（Ｑ）は、ポインタ論理回路５３０（図２９）からのＰｕｓｈ　Ｓｅｌｅｃｔ信号が、妥当性ビット・シンクロナイザが次の記号　－　ＳＫＩＰ記号でないならば　－　の受信に対して関連するＦＩＦＯのレジスタ５２０を選択するときに（論理“１”レベルに）セットされる。

【０１５６】遅延素子５４１ａ及びＯＲゲート５４１ｂは、通常設計のパルス引伸し回路を形成すべく動作し、フリップフロップ５４２のＳｅｔ入力での信号が少なくとも１クロック期間の持続時間を有することを確実にする。その場合でありかつ（ルータに対して）局所的なＲｃｖ　Ｃｌｋという知識及び受信したＴ__Ｃｌｋ信号が、もし同一でないならば、類似な周波数を有することを想定すると、ＲｃｖＣｌｋの少なくとも一つのアクティブ・トランザクションが、フリップフロップの出力（Ｑ）をセットすることによってフリップフロップ５４２に引伸ばされた信号を記録させるということが明らかになる（以下の、真理値表を参照）。D型フリップフロップ５４３は、同期の追加段階として作用し、局所Ｒｅｃ　Ｃｌｋに関してＶ出力で安定レベルを確実にする。Ｐｕｌｌ　Ｓｅｌｅｃｔ信号、引きポインタ５３２の復号は、フリップフロップ５４２のイネーブル入力に接続し、関連レジスタ５２０が読取られたときにＰｕｌｌ信号（ｓｙｎｃＦＩＦＯ制御装置５３４からの周期パルス）に妥当性シンクロナイザ５２２上の妥当性ビットをクリアさせる。

【０１５７】

【表４】

表　４

真理値表

セット	Ｒｓｔ	イネーブル	O_n	O_{n+1}
1	X	X	X	1
0	X	0	0	0
0	X	0	1	1
0	1	1	X	0
0	0	1	1	1
0	0	1	0	0

【０１５８】要約すると、妥当性シンクロナイザ５２２は、有効記号であるとその記号を識別するために記号がＦＩＦＯ５１８のレジスタ５２０に装填されるときに“妥当性”（Ｖ）信号をアサートすべく動作する。他方、記号がＳＫＩＰ記号であったならば、ＳＫＩＰ検査論理回路５４０の出力は、ＬＯＷに行き、フリップフロップ５４１（即ち、データ（Ｑ）出力）をゼロのままにさせて、関連記号が有効でないことを示し、かつ無視されるべきある。図２９を続けると、入力レジスタ５１６の内容は、ＳＫＩＰ検査論理回路５４０にも印加される。ＳＫＩＰ指令記号の受信は、ＳＫＩＰ制御論理回路５４０によって検出されたときには、押しポインタ論理回路５３０の動作を禁止（抑制）すべく動作し、かつＴ__Ｃｌｋの一付随クロック周期に対してクロック同期ＦＩＦＯ５１８にその記号を装填することを排除する。ＳＫＩＰ指令記号の受信は、押しポインタ５３０を先に進めないかまたは妥当性ビットＶをセットさせず、実質的に、押しサイドによるＳＫＩＰ記号の受信のＦＩＦＯ無知の引きサイドを保持する。レジスタ・パイプライン５２６から渡された入力データ／指令記号は、入力ストリームの指令記号が復号されかつＦＩＦＯ制御論理回路５４６を制御するために用いられる。エラスティックＦＩＦＯ5060 を動作することに加えて、ＦＩＦＯ制御論理回路５４６は、クロスバー論理回路５００を介してポート入力5020 から記号を受信するポート出力５０４に対して必要なハンドシェーク信号を生成すべく動作する。

【０１５９】指令／データ記号は、リンク・レベル“キープアライブ(keep-alive)”プロトコル（以下に説明する）、メッセージ・パケット終了検査、等を含んでい

る、リンク・レベル及びパケット・プロトコルを確認すべく動作するプロトコル及びパケット検査論理回路５５０にも印加される。（見出されたときに、記号ストリームから抽出される）指令記号でない、即ち、データ記号である、それらの記号は、アクセスされたときに、それからクロスバー論理回路５００に伝達される、エラスティックＦＩＦＯ5060に渡されかつ記憶される。また、メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎ ＩＤは、ターゲット・ポート選択論理回路５６０にも伝達される。ターゲット・ポート選択論理回路５６０は、メッセージが送信のために送られる、ポート出力５０４の“ターゲット・ポート”アドレスを、受信したＤｅｓｔｉｎａｔｉｏｎ ＩＤ及びルータの構成レジスタのある情報から決定すべく動作する。ターゲット・ポート選択論理回路５６０は、適切なクロス接続を行うためにクロスバー論理回路５００に印加されかつそれによって用いられる３ビット・コードを発生する。しかしながら、選択ポート出力５０４は、ポート入力5020からメッセージ・パケットを受信すべく“イネーブル”されなければならない。この目的のために、ポート入力5020は、どのポート出力５０４がポート入力5020からメッセージ・パケットを受信すべく許可されるかという情報を含んでいる、６ビット・ポート・イネーブル・レジスタ５６２を含む。ポート・イネーブル・レジスタ５６２の各ビット位置は、一つのポート出力５０４に対応しており、かつ特定ビット位置の状態により、対応ポート出力は、ポート入力からそれに送られるメッセージ・トラフィックを有すべく“イネーブル”されうるか、または“ディスエーブル”されて、ポート入力5020からそれに送られるメッセージ・トラフィックを排除する。例えば、ポート入力5020がターゲット・ポート選択論理回路に宛先ポートとしてポート出力5044を識別させる宛先情報を有しているメッセージ・パケットを受信し始めると想定する。しかしながら、ポート・イネーブル・レジスタ５６２の状態は、ポート出力5044がポート入力5020からメッセージ・トラフィックを受信すべく許可されないようであると更に想定する。この場合には、ポート・イネーブル・レジスタ５６２の内容は、ターゲット・ポート選択論理回路５０６によって発生された選択情報がクロスバー論理回路５００に印加されることを禁止すべく動作する。それよりも、パケットは、降下され、かつルータ１４Ａが、パケットが受信されたポートに対して許可されないポート向けのパケットを受信したということを示すべく誤り信号が生成される。誤りは、ＯＬＡＰ２８５’を介してＭＰ１８に知らされる（図２７）。

【０１６０】従って、ポート・イネーブル機能（特徴）は、ルータ１４を通るある一定のルーティング経路を選択的に防止すべく動作する。この機能は、デッドロック状態を防止する重要な機構でありうる。デッドロック状態は、メッセージを伝達すべく用いるネットワークが、ルーティング装置及び相互接続リンクによって形成された“ルーティング・ループ(routing loops)”を含むときに発生する。別のメッセージが既にそのポートから送られる処理中であるので、一つのルーティング装置で受信したメッセージが特定のポートから送られることからブロックされることが発生する。しかしながら、次いで、他のメッセージも第３のメッセージにより別のルーティング装置でブロックされる、等。全てのメッセージは、それぞれ環状ループでブロックされる。ループの各メッセージがループの別のメッセージによってブロックされ、かつループの別のメッセージをブロックしているので、何も移動しない；メッセージは、デッドロックされる。適切な設計なしで、大きなルーティング・ネットワークは、一群のメッセージ・パケットのそれぞれが通信リンクへのアクセスを獲得する前に別のものが進行するのを待たなければならないような環状（循環）依存性(circular dependencies)のために、通信ネットワークを通して更なる前進を行うことができないメッセージ・パケットのグループを結果として生ずるデッドロックに対する多数の環境の可能性を生じさせることができる。ルータを通るある一定の通信経路をディスエーブルできることにより、可能なルーティング・ループを除去することができ、それにより、デッドロックが発生する可能性を除去することができる。

【０１６１】もちろん、ルーティング・ループに対抗する防御の最初のライン及びデッドロックの可能性は、適切なルーティング情報が、入力メッセージ・パケットがルーティング・ループの部分でありうるルータ１４のポートから送られないようにターゲット・ポート・アドレスを選択するために用いられることを確実にすることである。しかし、ポート・イネーブル・レジスタによって達成されるような、ルータ１４を通るある一定のルーティング経路をディスエーブルする能力（機能）は、ルーティングまたは他の誤りがデッドロック状態を結果として生じないことを確実にする。この概念の実施は、以下により詳細に説明される。再度、図２９を続けると、入力メッセージ・パケットのヘッダが受信されると、Ｄｅｓｔｉｎａｔｉｏｎ ＩＤｓは、ターゲット・ポート選択論理回路５６０に逐次的に渡されかつ先着順にそこで試験される。ターゲット・ポート選択論理回路５６０は、指定ポート出力を識別する、ターゲット・ポート・アドレスを生ずる。上記したように、選択ポート出力５０４がポート入力に対してイネーブルされることを条件として － そのアドレスは、メッセージ・パケットを受信するエラスティックＦＩＦＯ５０６の出力を適切なポート出力に伝達する適切なクロスバー選択を行うためにクロスバー論理回路５００に印加される。（ルーティング１４がＣＰＵｓ１２への直接的なＴＮｅｔ接続を有するものであり、かつデュプレックス・モードで動作し

ているならば、ＣＰＵｓに向かう入力メッセージ・パケットは、メッセージ・パケットを同時に両方のポート出力５０４_4及び５０４_5にルーティングすることによりクロスバー論理回路装置によって複写される。）

ターゲット・ポート選択論理回路５６０は、図３１により詳細に示されており、かつ入力パケットの３バイトＤｅｓｔｉｎａｔｉｏｎ　ＩＤをポート出力５０２のエラスティックＦＩＦＯｓ５０６（図２７～図２９）から受信する、宛先レジスタ５７０を含んで示されている。Ｄｅｓｔｉｎａｔｉｏｎ（宛先）ＩＤは、図５（ｂ）に関して上述した３つのフィールドを含む：Ｒｅｇｉｏｎ（領域）ＩＤ，Ｄｅｖｉｃｅ（装置）ＩＤ，及び経路選択ビット（Ｐ）を含んでいる１ビット・フィールド。Ｒｅｇｉｏｎ（領域）ＩＤは、名前が示唆するように、領域による宛先を識別し、Ｄｅｖｉｃｅ（装置）ＩＤは、その領域内の特定の装置を表わす。経路選択ビット（Ｐ）は、経路（ＸまたはＹ）のどちらが二つのサブプロセッシング装置をアクセスするのために用いられるべきかを識別する。

【０１６２】ルータ１４は、例えば、大規模な並列処理アーキテクチャに対して、大きくて、用途が広いルーティング・ネットワークを構築する能力（機能）を供給する。ルータは、制御論理回路５０９に含まれるある一定のルータの構成レジスタにセットされた情報によりネットワークにおけるそれらの位置（即ち、レベル）によって構成される。これらの構成レジスタは、上部領域ＩＤレジスタ５０９_a、下部領域ＩＤ５０９_b、ＨｉＬｏレジスタ５０９_c、デフォルト・ポート・レジスタ５０９_d、クロスーリンク・ポート・レジスタ５０９_e、デフォルト・レジスタへの経路５０９_f、装置ＩＤ比較レジスタ５０９_g、及びサイド・レジスタ５０９_hとして図３１に示される。二つの追加構成レジスタは、装置位置としてかつレジスタ５０９_j及び５０９_kをそれぞれ伴って図３３に示される。これら種々の構成レジスタの内容は、Ｄｅｖｉｃｅ　ＩＤ及びメッセージ・パケットの付随経路選択ビット（Ｐ）と一緒に、メッセージ・パケットがクロスバー論理回路５００を通して送られるポート出力５０４の選択を決定する。ルータのレベルは、部分的に、Ｄｅｓｔｉｎａｔｉｏｎ　ＩＤのどの部分がターゲット・ポートの選択に用いられるか、及びアルゴリズム・アドレス選択を用いることができるか否かを決定する。この目的のために、Ｒｅｇｉｏｎ　ＩＤは、二つの重複１０ビット・レベル識別に更に分割される。Ｒｅｇｉｏｎ　ＩＤの内容の上位１０ビットは、上部レベルとして画定され、Ｒｅｇｉｏｎ　ＩＤの下位１０ビットは、下部レベルを指定する。両方のレベル識別は、マルチプレクサ５７２の二つの１０ビット入力の対応するものに印加される。マルチプレクサ５７２は、ルータのレベル（上部または下部）を識別し、かつ選択した１０ビットをアドレスとしてルーティング表５８４に供給するＨｉＬｏレジスタ５０９_cの内容に応じて二つの１０ビット入力の一つを選択する。

【０１６３】図２９及び図３１は、それぞれがそれら自身、個別のターゲット・ポート選択論理回路５６０、及びルーティング表５８４を有しているようにポート入力５０２を示す。しかしながら、空間を最小にするために、単一ルーティング表を全ての６つのポート入力５０２のターゲット・ポート選択論理回路によって共有することができることは、当業者には明らかであろう。マルチプレクサ５７２の出力は、通常のラウンド・ロビン・アービトレーション方法を用いて、アービトレーティド・ベイシスで(arbitrated basis)、（状態及び制御論理回路５０９に含まれうる）ルーティング表５８４にそれ自身マルチプレクスされうる。ルーティング表のアクセスの結果は、戻されかつマルチプレクサ５８６の入力に印加される。簡略化のために、このアービトレーション及びマルチプレキシングは、図３１に示さない。また、Ｒｅｇｉｏｎ　ＩＤの４つの上位ビットは、それらが上部領域ＩＤレジスタ５０９_aの内容と比較される４ビット比較回路５７４にも印加される。Ｒｅｇｉｏｎ　ＩＤの下位１０ビットは、それらが下部領域ＩＤレジスタ５０９_bの内容と比較される比較回路５７８に結合される。

【０１６４】例えば、ランダム・アクセス・メモリの形でだりうる、ルーティング表５８４は、複数の３ビット・ターゲット・ポート識別を記憶すべく動作する。Ｒｅｇｉｏｎ　ＩＤの一つのまたは他の１０ビット部分によってアドレスされたときに、ターゲット・ポート選択情報の３つのビットは、マルチプレクサ５８６の一つの入力に結合される；マルチプレクサ５８６の他の入力は、デフォルト・ポート・レジスタ５０９_dの３ビット内容を受信する。マルチプレクサ５８６によるルーティング表５８４の出力の選択は、（ルータが上部レベル・ルータであることを示している）ＯＮＥのときに、ＨｉＬｏレジスタ５０９_cの内容、または（宛先がこの“低レベル・ルータ”と同じ“高領域(high region)”であることを示している）比較回路５７４による上部Ｒｅｇｉｏｎ　ＩＤレジスタ５０９_aの内容とＲｅｇｉｏｎ　ＩＤの４つのＭＳＢｓとの間の成功した比較(successful compare)のいずれかによって行われる。それら条件のいずれも満足されないならば、マルチプレクサ５８６は、ターゲット・ポート識別としてデフォルト・ポート・レジスタ５０９_dの（３ビット）内容を代わりに選択する。ルーティング表５８４は、どんなサイズのものでもありうる。しかしながら、当業者に明らかなように、ルーティング表５８４のサイズは、ルータが用いられるシステムのアドレス可能素子の数、及び表に対して利用可能な空間のようなファクター（因子）によって指図される。ターゲット・ポート選択論理回路５６０は、ルーティング表において空間を確保するために、要求されたときに

表ルックアップ技術の使用、または要求されなかったときにアルゴリズム的ルーティングを組み合わせることにより新規な妥協（コンプロミス）を実施する。この組合せは、入力メッセージ・パケットを６つの利用可能なポートの一つへ渡し、かつそれから送信させ、及び非常に多種多用なルーティング能力を供給する。

【０１６５】マルチプレクサ５８６によって選択された３ビット・ターゲット・ポート識別は、マルチプレクサ５８６の出力とクロスーリンク・ポート・レジスタ５０９$_e$の３ビット内容との間を選択する更なるマルチプレクサ５９０の一つの（３ビット）入力に伝達される。二つの値のいずれが選択されるかは、入力メッセージの経路選択ビット（Ｐ）の状態によって示されるように最終宛先のサイド（即ち、ＸまたはＹ）によって決定される。入力メッセージ・パケットの経路選択ビット（Ｐ）は、その出力がマルチプレクサ５９０によって行われる選択に影響を与えるサイド・レジスタ５０９$_h$ の内容と比較される。ルータが、メッセージ・パケットが向けられたもの同じサイド（ＸまたはＹ）上でないならば、コンパレータ５９２の出力は、クロスーリンク・ポート・レジスタ５０９$_e$ の内容の選択に影響を与える。これは、ルータを含んでいるＸまたはＹサイドからメッセージ・パケットの宛先を含んでいる他のサイドにメッセージ・パケット直接的または間接的（即ち、別のルータを通す）に送るそのポート出力５０４にメッセージ・パケットを送る。マルチプレクサ５９０によって行われる選択は、その選択入力がＡＮＤゲート論理回路５９６の出力を受信するマルチプレクサ５９４の入力に印加される。マルチプレクサ５９４は、マルチプレクサ５９０及びマルチプレクサ５９８によって供給されるポート・アドレス間を選択する。次いで、マルチプレクサ５９８は、アルゴリズム・ルーティング論理回路６００の出力とデフォルト・ポート・レジスタ５０９$_d$ の内容との間を選択する。この選択は、Ｄｅｖｉｃｅ　ＩＤ（構成）レジスタ５０９$_g$ の内容及び入力メッセージのＤｅｖｉｃｅ　ＩＤの６ビットの選択部分を受信する選択及び比較回路６０１によって行われる。特に示されていないのは、アルゴリズム・ルーティング論理回路６００（図３３）のそれぞれ装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$ も、選択及び比較回路６０１に印加されるということである。装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$ に含まれる値は、アルゴリズム・ルーティング技術で用いられないＤｅｖｉｃｅ　ＩＤの高位ビットだけがＤｅｖｉｃｅ　ＩＤレジスタ５０９$_g$ の内容と比較されるようにメッセージのＤｅｖｉｃｅ　ＩＤビットをマスクすべく動作する。

【０１６６】メッセージのＲｅｇｉｏｎ　ＩＤの選択された（マスクされた）ビットとＤｅｖｉｃｅ　ＩＤレジスタ５０９$_g$ の内容との間のマッチは、可能なターゲット・アドレスとしてマルチプレクサ５９８でアルゴリズム・ルータ６００の結果を選択することを結果として生ずる。例えば、Ｒｅｇｉｏｎ　ＩＤが“ａｂｃｄｅｆ”（ａが高位ビットである）であり、かつ装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$ に含まれる値がビット“ｄｅｆ”がアルゴリズム処理に用いられるのようなものであるならば、Ｒｅｇｉｏｎ　ＩＤのビット“ａｂｃ”は、選択及び比較回路６０１によってＤｅｖｉｃｅ　ＩＤレジスタ５０９$_g$ の内容と比較される。逆に、ビット“ｃｄｅｆ”がアルゴリズム・ルーティングに用いられるならば、ビット“ａｂ”だけがＤｅｖｉｃｅ　ＩＤレジスタ５０９$_g$ の内容と比較される。また、メッセージのＤｅｖｉｃｅ　ＩＤのどのビットがアルゴリズム・ルーティングに含まれるかまたは含まれないかは、図３３に関して以下に説明されるよに装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$ によって決定される。その動作が以下により完全に説明される、アルゴリズム・ルーティング論理回路６００は、デフォルト・レジスタ５０９$_d$ の内容またはルーティング表５８４によって供給されるターゲット・ポート識別の代わりに選択されうる３ビット・ターゲット・ポート識別をそれから生ずるべく装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$ （明確さの理由により図３１には示されていない、図３３参照）によって供給される６ビットＤｅｖｉｃｅ　ＩＤ及び情報を受信する。アルゴリズム・ルーティング論理回路６００は、ルータが低レベル・ルータとして構成されるならば用いられる。

【０１６７】マルチプレクサ５９４によって行われた選択は、（デフォルト・レジスタ５０９$_f$ への経路の内容の状態により）その選択、またはデフォルト・ポート・レジスタの３ビット内容を選択処理の最終段階、検査論理回路６０２に渡す最終マルチプレクサ５９９に印加される。検査論理回路６０２は、ターゲット・ポート選択決定のプロダクト（結果）、マルチプレクサ５９９の出力によって識別されるポート出力の状態を検査すべく動作する。例えば、ターゲット・ポート識別は、有効でなければならない（即ち、６または７でない）。他の検査も行われ、その一つは、識別されたポート出力が、上述したようなアクセスをシークしている特定のポート入力に対して“イネーブル”されなければならないということである。ルーティング・ループを生成することができ、かつ次いで可能なデッドロック状態を発生させることを結果として生ずる、誤りに対するバックアップとして用いられるのは、この後者の検査である。検査論理回路６０２は、図３１に示すように、６つのポート出力５０２のそれぞれのポート・イネーブル・レジスタ５６２の内容を受信する。示したように、各ポート・イネーブル・レジスタ５６２の内容は、各入力ポート５０２に対して、どの出力ポート５０４が入力メッセージを送ることができ、そして勿論、どれができないかを識別する。それゆえに、例えば、ポート０がメッセージはポート３

からの送信のために送られるということを示しているＤｅｓｔｉｎａｔｉｏｎＩＤを含んでいるメッセージ・トラフィックを受信したならば、選択論理回路５６０は、ポート３としてターゲット・ポートを識別する３ビット量(3-bit quantity)を生じ、かつその量を検査論理回路６０２に印加する。更に、ポート３からのメッセージ・トラフィック送信がポート０で受信した入力メッセージ・トラフィックに対して許容されないということになったならば、ポート０に対するポート・イネーブル・レジスタ５８９の内容は、クロスバー論理回路５００へのターゲット・ポート・アドレスの伝達（通信）をブロックする。メッセージは、クロスバー論理回路５００の存在しない出力にその代わり送られ、かつ実質的に廃棄され、ＭＰシステム１８に知らせるべく誤り信号が生成される。

【０１６８】他方、ポート３がポート０から送られたメッセージ・トラフィックに対してイネーブルされるならば、検査論理回路６０２は、選択論理回路５６０によって発生されたターゲット・ポート識別をクロスバー論理回路５００に渡し、メッセージをポート３に送らせる。検査論理回路６０２は、通常設計のものであり、例えば、通常のファッションで行われるべき検査及び決定を実施すべく構成された組合せ論理回路を含んでいる。上部及び下部レベルの概念的階層が可視化されたことは、少なくとも部分的にターゲット・ポート選択論理回路のコンポーネント・カウント、及びルーティング表５８４のサイズを制限する理由のためである。そして、ルータ１４が上部または下部レベル・ルータを指定しうるし、かつサブプロセッシングシステム１０Ａ，１０Ｂの一つまたは他の一つに配置されうるということは、その階層によるものである。ルータが上部レベルまたは下部レベル・ルータであるかは、ルーティング表５８４をアドレス指定するために入力メッセージのＲｅｇｉｏｎ　ＩＤのどの部分が用いられるかも画定する、制御論理回路５０９のその種々の構成レジスタに書込まれた情報によって決定されるそのルータの構成による。

【０１６９】これらの概念を考えに入れて、図３２は、適切なポート出力へのクロスバー論理回路５００を通る入力メッセージ・パケットの経路を選択するために用いられる最終ターゲット・ポート・アドレスを選択すべく用いられる決定チャート６０４を示す。決定チャート６０４は、入力メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎ　ＩＤ（及び経路選択ビットＰ）、及びその構成レジスタ（即ち、図３１に示すレジスタ５０９$_a$，．．．，５０９$_h$）の内容によって指定されたように　－　そのルータの構成に基づいて行われた決定を示す。図３２が示すように、全ての決定をオーバーライドする(overriding)ことは、デフォルト・レジスタ５０９$_f$ への経路の内容である：　デフォルト・レジスタ５０９$_d$ の内容を選択すべくセットされたならば、全ての他の情報（Ｄｅｓｔｉｎａｔｉｏｎ　ＩＤ、経路選択ビットＰ、他の構成レジスタの内容、等）は、不必要になる（余計になる）。上述したように、各ルータは、上部または下部レベル・ルータのいずれかとして構成される。ルータ・レベルは、Ｄｅｓｔｉｎａｔｉｏｎ　ＩＤのどのビットがルーティング表５８４をアドレス指定すべく用いられか、かつアルゴリズム・ルーティングが用いられるべきか否かを決定する。（ＨｉＬｏレジスタ５０９$_c$の内容によってそのように識別される）高レベル・ルータは、ルーティング表、クロスーリンク・アドレス、またはデフォルト・アドレスのいずれかを用いる。低レベル・ルータ（ＨｉＬｏレジスタ５０９$_c$ がＺＥＲＯを含む）は、表ベース、デフォルト、クロスーリンク、及びアルゴリズム・ルーティングを用いる。

【０１７０】一般に、高レベル・ルータであるべく構成されたルータは、多数のルータ１４を備えておりかつ多数のＣＰＵｓ１２及びＩ／Ｏ装置を互いに伝達（通信）しているＴＮｅｔリンクＬを相互接続しているネットワーク“雲(clouds)”（任意ネットワーク）を相互接続すべく用いられ、大規模な並列処理（ＭＰＰ）システムを形成している。他のそのようなＭＰＰシステムが存在しうるし、かつそれは、一つのＭＰＰシステムのそのようなネットワーク雲を他のＭＰＰシステムに相互接続するために主に用いられる高レベル・ルータとして構成されるそれらのルータである。図２７～図２９をちょっと参照すると、入力メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎ　ＩＤは、特定のポートの入力論理回路５０２によって受信されたときには、エラスティックＦＩＦＯ５０６に、かつエラスティックＦＩＦＯ５０６からそれが捕獲されるターゲット・ポート選択論理回路５６０（図３１）のレジスタ５７０に伝達される。メッセージ・パケットのＤｅｓｔｉｎａｔｉｏｎＩＤがそのように捕獲されるとすぐに、選択処理が始まり、適切な出力ポートに　－　出力ポートがイネーブルされるということを想定して、両方とも一般的に、かつメッセージ・パケットを受信している特定の入力ポートに対して、クロスバー論理回路を通してメッセージ・パケットを指向すべく用いられるターゲット・ポート・アドレスの発生に進む。

【０１７１】ここで図３３を参照すると、３つの一に対して８ビット(8-bit to one)のマルチプレクサ６２０，６２２，及び６２４を含んでいるアルゴリズム・ルーティング論理回路６００が、詳細に示されている。マルチプレクサ６２０，６２２，６２４のそれぞれの３つの選択入力（Ａ，Ｂ，Ｃ）は、３ビット装置位置レジスタ５０９$_j$ の内容、制御論理回路５０９に含まれる構成レジスタの別のものを受信する。各マルチプレクサ６２０，６２２，６２４の入力（０，１，．．．）は、Ｄｅｖｉｃｅ　ＩＤの６ビットの所定のものを受信する。装置位置レジスタ５０９$_j$ の内容は、以下に説明する、表４の

符号化によりターゲット・ポート・アドレスとして用いるべき３ビットの選択を制御する。ＭＵＸｅｓ６２０，６２２，６２４によって選択されるＤｅｖｉｃｅ　ＩＤの３ビットは、直接用いられない。それよりも、選択されたビットは、２ビット幅レジスタ５０９$_k$の内容によって調整される、二つの入力ＯＲゲート６２６及び３つのＡＮＤゲート６２８（６２８ａ，６２８ｂ，及び６２８ｃ）と、制御帯状態論理回路５０９（図５０９）に含まれる構成レジスタの別のものとを備えている組合せ論理回路に印加される。組合せ論理回路のプロダクト（生成物）は、３ビット・ターゲット・ポート識別である。幅フィールドは、ポート選択に用いる装置フィールド・ビットの数を指定する。０（ゼロ）の幅フィールド値は、全てのアルゴリズム的にアドレス指定された装置は、ポート０を通して接続することを表わす。３の幅フィールド値は、アルゴリズム的にアドレス指定された装置がいずれのポートにも接続することができることを表わす。

【０１７２】装置フィールド幅拡張レジスタ５０９$_k$の内容は、クロスバー論理回路５００に適用されるターゲット・ポート・アドレスを指定するために用いるビットの選択を特定する。位置及び幅ビットの値及び意味は、以下の表４及び表５に示される。

【０１７３】

【表５】

表　５

装置フィールド幅 拡張	装置の数 使用したＩＤビット
００	０
０１	１
１０	２
１１	３

【０１７４】図９は、入力メッセージ・パケットのＤｅｖｉｃｅ　ＩＤのどのビットがＭＵＸｅｓ６２０，６２２，６２４のそれぞれによって選択されるかを示す。それゆえに、例えば、０００の装置ビット位置レジスタ５０９$_j$における（２進）値は、ＭＵＸｅｓ６２０，６２２，及び６２４に入力メッセージ・パケットのＤｅｖｉｃｅ　ＩＤから、それぞれ、ビット２、１、及び０を選択させる。逆に、装置ビット位置レジスタ５０９$_j$の内容が２進１００であるならば、ビット５及び４だけが、Ｄｅｖｉｃｅ　ＩＤのＭＵＸｅｓ６２０及び６２２によってそれぞれ選択される；ＭＵＸ６２４の出力は、残りのビット位置に対してＺＥＲＯに強要される。装置ビット位置レジスタ５０９$_j$における１１０及び１１１（２進）の値は、ＭＵＸｅｓ６２０，６２２，及び６２４の出力をＺＥＲＯに強要させて、ターゲット・ポート０を選択する。ＭＵＸｅｓ６２０，６２２，及び６２４によってそのように選択されたビットは、装置フィールド幅拡張レジスタ５０９$_k$の内容により用いられる。それゆえに、図１０が示すように、００の幅値は、ＭＵＸｅｓ６２０，６２２，及び６２４からのビットのいずれをも選択せず、０００のターゲット・ポート・アドレスを強要する。逆に、装置フィールド幅拡張レジスタ５０９$_k$における１０の幅値は、ＭＵＸｅｓ６２０，６２２，及び６２４によって選択されたビットの二つを用いる。

【０１７５】結果は、多くの場合、制限された値のセットを有する、３ビット・ターゲット・ポート数である。効果は、レジスタ５０９$_k$の内容によって指定された幅を有するレジスタ５０９$_j$の内容によって指定されたビット位置で始まる、３ビット・フィールドを生成することである。図２７～図２８及び図３３にちょっと戻ると、出力ポート５０４は、メッセージ・パケットを順序付ける役割を果たす。一般に、メッセージ・パケットは、先着順でポート出力５０４によって受容されかつ送られる。しかしながら、一度メッセージ・パケットが特定のポート出力から送信されると、多数の他のものがそのポート出力へのアクセスを待って遅延（停滞）されうる。それゆえに、アービトレーション方法は、これらのパケット間で選択することが必要であろう。ラウンドロビン・アービトレーションのような、多数の通常のアービトレーション技術を用いることができる。しかしながら、好ましいアービトレーション技術は、１９９５年６月６日に出願され、かつ本出願の出願人にアサインされた、“Biased Routing Arbitration of Message traffic in Communication System（通信システムにおけるメッセージ・トラフィックのバイアスされたルーティング・アービトレーション）”という発明の名称であり、現在係属中の米国出願に開示されたものである。

【０１７６】簡単に、各ポート出力５０４は、自律アービター論理回路６３０を含む（図３４）。これらのアービター６３０は、ポート入力５０２のそれぞれからルーティング要求を取り、かつ使用されることが許可されるポート出力の合計帯域幅の割合いを表わすバイアス値を各ポート入力５０２が実質的に供給されるバイアシング技術に基づく順番でそのようなサービスを与える。このアービトレーション技術によれば、ポート出力５０４の一つに送られるべきメッセージ・トラフィックを有しているポート入力５０２は、アクセスに対するそれらの要求を知らせる。二つ以上のポート入力がアクセスをシークしているならば、要求されたポート出力は、それぞれのバイアス値を比較することによってポート入力をアー

ビトレートし、アクセスに対して一つ（例えば、最も高いバイアス値を有するポート入力）を選択する。アービトレーションに負けたポート入力は、次のアービトレーションの間にそれらの機会を増やすべくそれらの対応バイアス値が変更される；勝ったポート入力５０２もそのバイアス値が変更されるが、次のアービトレーションで勝つことの機会を減少するためである。ここで図３４を参照すると、それからメッセージ・パケットがルータ１４Ａ（図２７）によって送信されるポート出力５０４$_n$のブロック図が示されている。ポート出力５０４の基本コンポーネント及びそれらの機能は、：

－　入力ポートの間をアービトレートすべく動作し、パケットが出力ポートによって送信される順序を決定するアービター論理回路６３０。

【０１７７】－　プロトコル規則を維持しかつそれに従うために必要なときに指令記号を生成しかつ記号ストリームの中に（マルチプレクサ６３４を用いて）挿入すべく動作する指令記号発生器６３２。例えば、ルータ１４Ａが、受信素子が使用中なのでそれが送信することができないということを見出したときには、関連ポート出力５０４は、ＢＵＳＹ指令記号の受信に応じてメッセージ・パケット送信を停止することによって“バックプレッシャ”を課さなければならず、かつメッセージ・パケットの送信がＲＥＡＤＹ指令記号の受信によって示されるように再開することができるまでＦＩＬＬまたはＩＤＬＥ記号を挿入する。それが既に進行中のメッセージ・パケットを停止しなければならないならば、それは、ＦＩＬＬ記号を送る。代替的に、ＢＵＳＹ指令記号が受信されたときにルータ１４Ａのポートが活動停止状態（メッセージ・パケットが送られない）であるならば、それは、ＩＤＬＥし、かつＲＥＡＤＹ指令記号がＢＵＳＹ指令記号を早めに送ったシステム素子から受信されるまでメッセージ・パケットの開始を遅延する。ＦＩＬＬ記号は、指令記号発生器６３２によってポート出力５０４に供給される。プロトコルは、“キープアライブ”機構を実施するために出力論理回路も必要とする：ルータ１４Ａがまだ動作状態であることを受信素子に知らせるための記号の周期的送信（即ち、メッセージ・パケットがないときの、ＢＵＳＹ、ＩＤＬＥ）。キープアライブとして用いる記号の型は、そのときに存在しているい動作のモードによる。例えば、メッセージ・トラフィックがない期間中、ＲＥＡＤＹ記号が用いられかつ送信クロック、Ｔ＿Ｃｌｋの各クロック周期またはサイクルで周期的に送られる。代替的に、ポート出力がバックプレッシャを作用させたならば、ＢＵＳＹ記号は、送られる。規定された時間内に、記号をまったく受信しないと、アクションに対するルータ（またはインターフェイス装置）のＯＬＡＰを介してＭＰ１８にポストされる誤りを結果として生ずる。

【０１７８】脇道にそれて、ルータ１４によって観測されるこれらのプロトコル規則は、ＣＰＵｓ１２（即ち、インターフェイス装置２４）及びＩ／Ｏパケット・インターフェイス１７によっても観測されるということを理解すべきである。そして、ルータ１４ＡがＣＰＵｓ１２Ａ，１２Ｂと直接的に伝達すべくシステム１０（図１）にあり、かつデュプレックス・モードが用いられるときに、デュプレックス動作論理回路装置６３８は、ＣＰＵｓ１２Ａ，１２Ｂの一つにも接続される他のものにＣＰＵｓ１２Ａ，１２Ｂの一つに接続されるポート出力を協調すべく用いられる。ルータ１４Ａのポート出力５０４のそれぞれは、パケット順位付けの役割を果たす。一般に、パケットは、先着順に送られる。しかしながら一度パケットが送信されたならば、複数の他のものは、待たされ続けうる。ルータ入力論理回路５０２のそれぞれからルーティング要求を取り、かつ上記した出願係属中の出願において示される優先度スキームに基づいて適切な順序でサービスを要求している各入力ポートに出力ポートを与えることは、各ポート出力５０４のアービター論理回路６３０の機能である。ポート出力５０４の各アービター６３０は、それが要求を与えるときに全ての他のアービター６３０に信号を送る。

【０１７９】それは、二つのＣＰＵｓ１２から伝達された（デュプレックス動作における）同一記号のペアを受信する入力論理回路のクロック同期ＦＩＦＯｓ５１８である。各クロックｓｙｎｃ５１８ＦＩＦＯは、二つのＣＰＵｓ１２からの記号ストリーム間で発生する遅延原因スキューに対して調整することができる。ＭＰ１８にＣＰＵ１２への通信アクセスを供給したオンライン・アクセス・ポート（ＯＬＡＰ）がＭＣに含まれていたことが、上記ＣＰＵｓ１２の説明に関連して思い出される。ＭＰ１８は、ＣＰＵ１２にブート（開始）動作を終了させるべく小さなメモリ画像及びルーチンを構築するためにプロセッサ２０によって実行されうるＯＬＡＰ２８５に命令を書込むことができた。同様なアクセスがルータ１４へのＭＰ１８に供給される。図２７に少しの間戻ると、ルータ１４Ａは、ターゲット・ポート選択論理回路の上部及び下部領域レジスタ５０９ａ，５０９ｂ（図３１）、及びアルゴリズム・ルーティング論理回路６００（図３３）の装置ビット位置及び拡張レジスタ５０９$_j$，５０９$_k$のような、多数の構成レジスタを含むＯＬＡＰ２８５’を含んで示されている。ルータ１４Ａを含んでいるサブーシステム１０Ａの初期化の間中、ＯＬＡＰ２８５’に含まれる構成レジスタは、それを一つのマナーまたは別のマナーで動作させる構成をルータ１４Ａに供給すべく（ＯＬＡＰバス２８７’を介して）ＭＰ１８による情報で書込まれる。

【０１８０】しかしながら、ルータ１４Ａは、ＯＬＡＰ２８５’を通してＭＰへ情報（誤り表示、等）を渡しうる。例えば、ルータ１４Ａによって送られた各メッセージ・パケットは、上記したように、そのＣＲＣが検査さ

れる。パケットのＣＲＣが不良であるとルータ１４によって決定されたならば、ＴＰＢ記号でメッセージ・パケットにタグを付けることに加えて、ルータは、ＭＰ１８によって後で読取ることができるＯＬＡＰ２８５’に含まれる誤りレジスタ（図示省略）をセットすることによってＭＰ１８にフラグをたてる。それゆえに、システムは、この特徴（機能）を通して送信フォルトを知らせる手段を備えている。

【０１８１】クロッキング：明らかに、ＣＰＵｓ１２がデュプレックス・モードで同期的にマッチしたペアとして適切に動作されるならば、それらが用いるクロック信号は、同期されていなければならない。図３６は、クロック発生回路設計を示す。同期を維持するために各サブープロセッサ・システム１０Ａ／１０Ｂに一つのクロック発生器回路が存在する。参照番号６５０で一般に指定される、クロック発生器回路は、クリスタル発振器回路６５２ａ及び逓減係数８(divide-by-eight) の計数回路（カウンタ）６５２ｂを備える発振器回路６５２を含む。クリスタル発振器回路６５２ａは、２５／１６　５Ｍｈｚの周波数を有するマスタ・クロック（Ｍ＿ＣＬＫ）信号を生起すべく８によって分割される１２．５Ｍｈｚの周波数を有する周期的信号を生成する。Ｍ＿ＣＬＫ信号は、ＳＹＮＣ　ＣＬＫにも印加される。クロック発生器６５４に印加されて、Ｍ＿ＣＬＫ信号は、Ｍ＿ＣＬＫに対して全てが位相封じ込みされた、多数の５０Ｍｈｚクロック信号を生起すべく用いられる。これらの５０Ｍｈｚ信号は、クロック回路６５０を含んでいるサブープロセッサ・システム（例えば、１０Ａ）の種々の素子（例えば、ＣＰＵ１２、ルータ１４、等）に分配されかつ用いられる。

【０１８２】クロック発生器６５４は、Ｍ＿ＣＬＫ信号を受信しかつ、それ自身の位相ー封じ込みされた複製である、フィードバック・クロック信号と比較すべく接続された位相コンパレータ６６０を含んで示される。Ｍ＿ＣＬＫとフィードバック・クロック信号との間の位相差を表わすアナログ電圧（Ｖ）である、位相コンパレータ回路６６０の出力は、位相及び周波数の両方において、Ｍ＿ＣＬＫ信号に対してクロック発生器によって生成された５０Ｍｈｚ信号のロックを維持すべく電圧制御型クリスタル発振器（ＶＣＸＯ）６６２に印加される。位相コンパレータ６６０が、Ｍ＿ＣＬＫとフィードバック信号の間で所定の位相範囲よりも大きい位相差を検出したならば、それは、位相封じ込み(phase lock)の損失を示すべくＬＯＣＫ信号をディアサートする。ＶＣＸＯ６６２（図３６）は、厳しい許容誤差範囲(tight tolerances)内で動作すべく構成された１００Ｍｈｚ電圧制御型クリスタル発振器である。ＶＣＸＯ６６２の生成物は、５０Ｍｈｚ信号を生成すべく２で、かつＭ＿ＣＬＫ信号の複製、フィードバック信号を生成すべく６４で、ＶＣＸＯ６６２の出力をカウント・ダウン（分割）する同期カウンタに印加される。カウンタ６６３によって生成された５０Ｍｈｚクロック信号は、サブープロセッサ・システム中の必要な場所に分配される。

【０１８３】ここで、図３７を参照すると、周波数封じ込み動作用の一対のサブープロセッサ・システム１０Ａ，１０Ｂ（図１～図３）に対する同期クロック信号を生起するために用いられる二つのクロック回路６５０の相互接続及び使用を示す。図３７に示すように、サブープロセッサ・システム１０Ａ，１０Ｂの二つのＣＰＵｓ１２Ａ及び１２Ｂは、それぞれ、発振器回路６５２Ａ，６５２Ｂを含んでいる、クロック回路６５０Ａ及び６５０Ｂとして図３７に示す、クロック回路６５０を有する。しかしながら、ＣＰＵｓ１２の一つのクロック発振器６５２だけが、両方のＣＰＵｓ１２に対するＭ＿ＣＬＫ信号を生起するために用いられる。図３７は、両方のＣＰＵｓ１２のクロック発生器６５４Ａ及び６５４Ｂを駆動するために用いられるＣＰＵ１２Ａの発振回路６５２Ａを示す。ドライバ及び信号回線６６７は、サブープロセッサ・システム１０Ｂのクロック発生器６５４Ｂに発振器回路６５２Ａによって生起されたＭ＿ＣＬＫ信号を送付すべく二つのサブープロセッサ・システムを相互接続する。フォルト分離に対し、かつ信号の品質を維持するために、Ｍ＿ＣＬＫ信号は、個別ドライバ及びループバック接続６６８を通してサブープロセッサ・システム１０Ａのクロック発生器６５４Ａに送付される。ループバック接続６６８の理由は、信号相互接続６６７によって課せられた遅延によりクロック発生器６５４Ｂによって見られるものにおおよそし等しい遅延を発振器回路６５２Ａとクロック発生器６５４Ａとの間に課すことである。

【０１８４】簡略化のために、図３７に特に示されないのは、発振器回路６５２が発振器６５２Ａからのそれらをミラーする接続及びドライバを有するということである。それは、どの発振器回路６５２Ａ，６５２Ｂが二つのクロック発生器６５４Ａ，６５４Ｂを駆動する発振器であるかを制定するＣＰＵｓ１２Ａ，１２Ｂを接続するための用いるケーブルである。即ち、一つの方法(one way) で接続されて、ケーブル（図示省略）は、サブープロセッサ・システム１０Ａ，１０Ｂ間で図３７に図示した接続を制定する；別の方法で接続されて、接続は、同じであるが、発振器６５２Ｂが用いられる発振器である。図３７を続けると、サブープロセッサ・システム１０Ａの発振器回路６５２Ａによって生成されたＭ＿ＣＬＫ信号は、それらの対応ＳＹＮＣ　ＣＬＫ信号及びクロック発生器６５４Ａ，６５４Ｂによって生成された５０Ｍｈｚ信号から生起された種々の他のクロック信号のように、両方のサブープロセッサ・システム１０Ａ，１０Ｂによって用いられる。それにより、ペアになったサブープロセッサ・システム１０Ａ，１０Ｂのクロック信号は、デュプレックス・モードに必要な周波数封じ込み動

作に対して同期される。

【０１８５】クロック発生器６５４Ａ，６５４ＢのＶＣＸＯｓ６６２は、通常設計のものであり、かつ位相コンパレータ６６０からの印加されたアナログ電圧（Ｖ）が、制御限度の外側である（位相コンパレータ６６０から受信したクロック信号がひどく位相がずれていることを示している）ときでさえも所望の周波数を維持し続ける型のものである。これは、サブープロセッサ・システムがもはや周波数封じ込みされていないけれども、両方のクロック発生器６５４Ａ，６５４Ｂに、発振器回路６５２Ａの不適切な動作のフェース(face)において二つのサブープロセッサ・システム１０Ａ，１０Ｂにクロック信号を供給することを継続させる。クロック発生器回路６５４Ａ，６５４Ｂの（Ｍ__ＣＬＫが存在しかつその複製、フィードバック信号と同期(sync)であることを示している）位相コンパレータ６６０によってアサートされたＬＯＣＫ信号は、両方ともに誤り論理回路６７０Ａ，６７０Ｂに結合される。ＬＯＣＫ信号をアサートすることは、クロック発生器６５４によって生成された５０Ｍｈｚ信号が、Ｍ__ＣＬＫ信号に、位相及び周波数の両方において、同期されることを意味する。それゆえに、ＬＯＣＫ信号のいずれかがＺＥＲＯであった（即ち、ディアサートされた）ならば、誤り論理回路６７０は、どのクロック発生器がそのＬＯＣＫ信号をディアサートしたがを決定しかつＯＬＡＰ２８５を介してＭＰ１８に知らせる。両方のＬＯＣＫ信号がディアサートされたならば、ＣＰＵｓは、クロック発生器６５４Ａ，６５４Ｂを駆動している発振器回路６５２Ａが正しく動作していないことをそれから想定することができる。

【０１８６】－　一定比率クロッキング：上述したように、一対のデュプレックスされたＣＰＵｓ１２とルータ１４Ａ，１４Ｂ（図１～図３）の間の記号転送は、周波数封じ込みモードでそのように行われる；即ち、記号ストリームを付随し、かつ受信素子（ルータ１４、またはＣＰＵ１２）クロック同期ＦＩＦＯに記号をプッシュする（押す）ために用いられるクロック信号は、クロック同期ＦＩＦＯｓから記号をプルする（引く）ために用いられた受信素子のものに、位相でなければ、周波数において実質的に同一である。例えば、ルータ１４Ａから一対のデュプレックスされたＣＰＵｓ１２Ａ，１２Ｂに送られた記号を示す、図３５を参照すると、ルータ１４Ａで発生している（かつ受信クロック（Ｒｃｖ　Ｃｌｋ）としてＣＰＵｓ１２Ａ，１２Ｂで受信されるべき、記号ストリームを付随している、）クロック信号は、局所クロック（Ｌｏｃａｌ　Ｃｌｋ）に周波数において実質的に同一である。前者（ＲｃｖＣｌｋ）は、各ＣＰＵのクロック同期ＦＩＦＯｓ１２６に記号をプッシュするために用いられ、後者は、ＦＩＦＯｓから記号をプルするために用いられる。この技術は、同じ周波数のものであり、クロック信号（Ｔ__Ｃｌｋ／ＲｃｖＣｌｋ及びＬｏｃａｌ　Ｃｌｋ）に対してよく動作し、かつ偶然にもＴＮｅｔリンクＬ上の通信に用いられたクロック周波数である。しかしながら、送信媒体、即ち、ＴＮｅｔリンクＬの電気的または他の特性に従うために、その媒体にわたり記号を送信するために用いられたクロック信号の周波数が制限されが、受信エンティティが制限されないと仮定したならば、ここでＣＰＵｓ１２は、より高い周波数のクロック信号で動作することができる。そのようが情況において、それぞれのクロック同期ＦＩＦＯｓ１２６からプルされた記号に対して同期が二つのＣＰＵｓの間で維持されるということを確実にするために準備がなされなければならない。

【０１８７】ここで、一定比率クロッキング機構が、二つのクロック同期ＦＩＦＯｓ１２６の動作を制御するために用いられ、それらがＦＩＦＯｓにプッシュされるのと同じ速度で二つのＦＩＦＯｓから記号をプルするクロック信号を供給する。図３８を参照すると、参照番号７０で示された一定比率クロック制御機構が示されている。図３８が示すように、クロック同期ＦＩＦＯ制御機構７００は、その並列出力がＮ対１マルチプレクサ（ＭＵＸ）７０４に印加されるプル・セット可能なマルチ・ステージ直列シフト・レジスタ７０２を含む。直列シフト・レジスタ７０２は、シフト・レジスタのクロック（ＣＫ）入力に印加されるより速い（より高い周波数）局所クロック信号（Ｌｏｃａｋ　Ｃｌｋ）で動作される。１５ビット・バス７０１は、直列シフト・レジスタ７０２をプリセットすべくデータ入力（ＤＩ）にプリセット（PR_1）を運ぶ。　直列シフト・レジスタを形成しているステージ数は、わかるように、局所的に用いられるクロック信号の周波数に対する記号が伝達されかつクロック同期ＦＩＦＯｓ１２６にプッシュされるクロック信号の比率により、あらゆるものでありうる、ということが当業者には明らかであろう。ここでは、１５ステージが十分であると思われる。

【０１８８】ＭＵＸ７０４は、シフト・レジスタ７０２から１５並列データ出力（ＤＯ）の一つを選択すべく動作し、クロック同期ＦＩＦＯｓ１２６から記号をプルし、かつプル・ポインタ・カウンタ１３０を動作（更新）すべくＬｏｃａｌ　Ｃｌｋ信号として用いられる一定比率クロック制御機構の、出力として、ＭＵＸの入力（Ｉ）に印加される。選択された出力は、ＭＵＸの出力（Ｏ）からも結合されかつ直列シフト・レジスタのシフト・イン（ＳＩ）入力に印加される。選択は、４ビット・カウンタで実施されうる　－　サイクル長論理回路のデータ入力（ＤＩ）に印加された（４ビット）プリセット（PR_2）値でプリセット可能なサイクル長論理回路７０６によってなされる。サイクル長論理回路の４ビット出力は、ＭＵＸ７０４の選択（Ｃ）に印加される選択値を形成する。本質において、一定比率クロック制御は、所定の時期（期間）にわたりＲｃｖＣｌｋで同じ数

のクロック・エクスカージョン(excursions)を有している出力信号を生成すべく動作する。クロック同期ＦＩＦＯ１２６に記号をプッシュするために用いられるクロック信号に対するＣＰＵ１２のクロック信号の間のＮ：Ｍ（ここでＮ＞Ｍ）の比率が、Ｒｃｖ　Ｃｌｋであると想定すると、直列シフト・レジスタは、シフト・レジスタのＭステージが第１のディジタル状態（例えば、ＯＮＥ）を保持し、かつ他が別のディジタル状態（例えば、ＺＥＲＯ）を保持するようにプリセットされる。サイクル長論理回路は、（もちろん、最後または１５番目のステージが第１のステージへのフィードバックを形成するような、Ｍが１５であることを除き）Ｍステージを有するトランケートされた直列シフト・レジスタを、実質的に、生成する直列シフト・レジスタの出力を選択すべく値でプリセットされる。例は、これをよりクリアにする。

【０１８９】図３５をちょっと参照して、記号が５０Ｍｈｚクロックでルータ１４Ａから二つのデュプレックスされたＣＰＵｓ１２に送信されると想定する。それゆえに、記号は、５０Ｍｈｚ速度でＣＰＵｓのクロック同期ＦＩＦＯｓ１２６にプッシュされる。ＣＰＵｓのクロック信号が４０Ｍｈｚであると更に想定する。従って、Ｒｃｖ　Ｃｌｋ信号に対する局所クロック（８０Ｍｈｚ）の比率は、８：５である。直列シフト・レジスタは、１５ステージの最初または第１の８つが５つのＯＮＥｓと３つのＺＥＲＯｓを含むようなビット・パターンでプリセットされる。サイクル長論理回路は、ＭＵＸ７０４による直列シフト・レジスタの８番目のステージの選択を動作する値でプリセットされる。それゆえに、シフト・レジスタ及びサイクル長論理回路は、それぞれが１００ｎｓ（ナノ秒）期間である３つの“待ち(wait)”状態及び５つの“出る(out)”状態を、実質的に、含んでいる８つのステージを有する直列シフト・レジスタを、実質的に、生成する値が供給される。従って、クロック同期ＦＩＦＯｓ１２６から記号をプルするクロック信号を生成する、ＭＵＸ７０４の出力、Ｒｃｖ　Ｃｌｋは、各１００ｎｓ期間に対して、５つのクロック・パルスを含む。それゆえに、各１００ｎｓ期間に対して、５つの記号がクロック同期ＦＩＦＯｓ１２６にプッシュされ、かつ５つの記号がクロック同期ＦＩＦＯｓ１２６からプルされる。

【０１９０】この例は、図３９に記号的に示され、図４０に示されたタイミング図は、制御論理回路７００の動作を示す。各１００ｎｓ期間に対して、Ｒｃｖ　Ｃｌｋの５つのクロック・パルス（図４０において“ＩＮ”と示された）は、クロック同期ＦＩＦＯｓ１２６に記号をプッシュする。それと同じ１００ｎｓ期間中に、直列シフト・レジスタ７０２は、ＭＵＸ７０４によって選択されたステージ７１０を通して“０１１０１０１１”シーケンスを循環させ、Ｒｃｖ　Ｃｌｋ信号と同じ数のアクティブ・クロック・パルスを有するＬｏｃａｌ　Ｃｌｋ信号を生成する。シフト・レジスタ７０２のステージの数は、ここの示されたようなシステムにおける最も一般的なクロック・スピード差分を収容すべく変更されうるということが当業者には明らかである。好ましくは、シフト・レジスタ７０２は、示されたように、１５のステージを有し、比較的広い範囲のクロック比率をカバーするための能力（機能）を供給する。ここで理解できるように、この一定比率クロッキングのこの技術は、決して１クロック以上ずれない。更に、例えば、５つのクロックをカウントしかつ追加の記憶（例えば、同期ＦＩＦＯのサイズにおける増加）を必要としかつより待ち時間を課す３つのクロックを保持するよりも良好な実施である。

【０１９１】ここに示された一定比率クロック回路（図３８及び図３９）は、一周波数のクロック様式(regime)から異なる、より高い周波数のクロック様式にデータ素子を転送するために用いられる。クロック同期ＦＩＦＯの使用は、二つの異なるソースから同一の指令／データ記号のペアを受信すべく同期化された、デュプレックスド・モードで動作するときに、信号遅延の影響を補償するためにここで必要である。しかしながら、ここに開示された一定比率クロック回路は、クロック同期ＦＩＦＯの代わりに少なくとも二つのレジスタが存在する限り、二つの異なるクロック様式間でデータを伝達するために有用であるというが当業者に自明である。より高い周波数クロック様式からより低い周波数クロック様式にデータを転送することは、一定比率クロック回路７０２によって生起されたクロック信号の制御下で入力ステージまたはレジスタにデータ素子を転送すべく一定比率クロック回路７０２を用いる；より低いクロック様式のクロック信号は、二つ（または、ここでは、それ以上）の受信レジスタ・ステージの間でデータ素子を転送し、かつそれからデータ素子を除去するために用いられる。逆に、より低い周波数クロック様式からより高い周波数を有しているものに転送されたデータ素子は、ここに示されたように本質的に動作する。

【０１９２】この概念は、異なるクロック信号が用いられるところに用いることができる。例えば、マイクロプロセッサ技術においてよく知られるように、多くのマイクロプロセッサは、一周波数のクロック信号に応じて動作するマイクロプロセッサが、異なる、通常より低い周波数のクロック信号に応じて動作する同期装置（例えば、メモリ、または外部、システム・バス）と伝達するときに“待ち”状態を挿入すべく構成されている。一般に、そのようなマイクロプロセッサ／装置通信は、より遅いクロック信号がマイクロプロセッサ・クロック周波数の整数倍であるということが要求される。一定比率クロック制御回路７０２は、広い範囲の可能なクロック比率を供給できる。

【０１９３】Ｉ／Ｏパケット・インターフェイス：サブープロセッサ・システムが局所Ｉ／Ｏを有することを必要としないように他のサブープロセッサ・システムのＩ／Ｏが利用可能であるということが考えられるけれども、サブープロセッサ・システム１０Ａ，１０Ｂ、等のそれぞれは、種々の周辺装置で実施される、入力／出力機能を有する。あらゆる場合において、局所Ｉ／Ｏが供給されたならば、周辺装置及び／又はＭＰ１８は、Ｉ／Ｏパケット・インターフェイス１６を介して伝達（通信）する。Ｉ／Ｏパケット・インターフェイス１６は、装着されたＩ／Ｏ装置に対してより互換性があるかまたはネーティブな形にそれがＴＮｅｔリンクＬから受信した入力メッセージ・パケットを変換すべく動作する；次いで、Ｉ／Ｏパケット・インターフェイス１６は、反対方向にも変換して、装着されたＩ／Ｏ装置から“ネーティブＩ／Ｏ”（ＮＩＯ）を受信し、上述した８Ｂ－９Ｂフォーマットのデータにバイトを符号化し（上記、表１参照）、かつデータを宛先に送るために必要なパケットを形成する。更に、特定のＩ／Ｏ装置（例えば、信号回線）に対して最も普通の方法でアサートされる、Ｉ／Ｏ装置からの割込みは、Ｉ／Ｏパケット・インターフェイス装置１６によって受信され、かつそれが上述したように処理される、割込みが意図したＣＰＵ１２に送られる割込みパケットを形成するために用いられる。それゆえに、ＮＩＯバス上の装置は、読取りをし、書込みをし、かつＣＰＵ１２のメモリ２８にＴＮｅｔリンクＬ及びルータ１４を通して透過的に渡されたデータ／制御情報で慣例的にメッセージ・パケットを介して割込みを発行する。

【０１９４】そして、Ｉ／Ｏパケット・インターフェイス１６が、Ｉ／Ｏ装置の一つ、ＭＰ１８として、それに接続されうるけれども、Ｉ／Ｏパケット・インターフェイス１６は、構成情報を、ＯＬＡＰバスを介して、受信するＭＣ２６（図１７Ｂ）にに含まれ（ＯＬＡＰ２８５）かつルータ１４に含まれた（ＯＬＡＰ　２８５’；図２７）ようなＯＬＡＰも含む。

【０１９５】オンーライン・アクセス・ポート：ＭＰ１８は、（ＩＥＥＥ１１４９．１－１９９０、５月、１９９０年、ＳＨ１３１４４、Institute of Electrical and Electronic Engineerings, 345 East47th Street, New York, NY 10017 に基づく）ＩＥＥＥ標準１１４９．１に準拠するインターフェイス信号でインターフェイス装置２４、メモリ・コントローラ（ＭＣ）２６、ルータ１４、及びＩ／Ｏパケット・インターフェイスに接続する。ＯＬＡＰ２５８は、そのＩＥＥＥ標準を実施し、かつＯＬＡＰ２５８の構成及び動作は、それがどの素子（例えば、ルータ１４、インターフェイス装置２４、等）を用いたかに関係なく本質的に同じである。図４１は、ＩＥＥＥ１１４９．１標準インターフェイスを実施しているＯＬＡＰ２５８の一般構造を図式的に示す。ＯＬＡＰは、ここに説明した素子のあるものを実施するために用いられる各集積回路チップ上で形成されるのが好ましい。例えば、各インターフェイス装置２４、メモリ・コントローラ２６、及びルータ１４は、ＯＬＡＰも含むアプリケーション指定集積回路（ＡＳＩＣ）によって実施され、ＡＳＩＣの回路素子へのアクセスをＭＰ１８に供給する。それゆえに、図４１に示したＯＬＡＰ１５８の説明は、インターフェイス装置２４、ＭＣ２６、及びシステムのルータ１４に関連するＯＬＡＰを説明する。

【０１９６】図４１に示すように、直列バス１９Ａは、４つの１ビット信号回線を備えている：ＯＬＡＰ２５８へ周期的クロック信号を運ぶ試験クロック（ＴＣＫ）信号回線；２状態指令信号を伝達する試験指令（ＴＣＭ）信号回線、ＯＬＡＰへデータを運ぶ試験データ・イン（ＴＤＩ）信号回線；及びＯＬＡＰからデータを伝達する試験データ・アウト（ＴＤＯ）信号回線。これらの信号は、ＩＥＥＥ１１４９．１標準の要求事項に従っている。ＯＬＡＰ２５８は、直列バス１９ＡのＴＣＫ及びＴＣＭ回線上で受信したクロック及び指令信号に応じてＯＬＡＰの動作を制御する４ビット状態マシンを含む。ＯＬＡＰ２５８によって受信されたデータ（及び／又は命令）は、１６ビット命令レジスタ（ＩＲ）８０２及び／又は３２ビット・データ・レジスタ（ＤＲ）１０４によって記憶される；データは、ＤＲ８０４だけが関連論理回路（例えば、ルータ１４）からのデータで装填することができるということを除き、ＩＲ、ＤＲレジスタのいずれかから伝達されうる。ＯＬＡＰ２５８に関連するが、その一部でないものは、ＭＰ１８（ＯＬＡＰ２５８を介して）及びＯＬＡＰ２５８が関連される論理回路の両方によってアクセスすることができる６４の３２ビット・レジスタまでを含むレジスタ・ファイルの形の構成レジスタ８０６である。例えば、構成レジスタ８０６のレジスタのあるものは、ルータ１４の制御及び状態論理回路５０９（図２７）を形成する。構成レジスタ８０６は、ＩＲ８０２によって最初に供給される１０ビット命令によって命令される（３２ビット）位置（即ち、６４の利用可能な３２ビット・アドレスの選択されたもの）でＤＲ８０４から書込まれる。構成レジスタ８０６を装填するための命令は、命令復号論理回路８１０によって復号される４ビット部分を含み、指令発生器８１２に印加された合成復号は、読取りまたは書込み動作を識別する。動作の目的、即ち、読取られるかまたは書込まれるべき、構成レジスタ８０６を構成している６４レジスタの一つは、アドレス復号論理回路８１４によって復号される６ビット・アドレスによって識別される。指令発生器８１２は、状態マシン８００の状態も受信する。それゆえに、状態マシン８００によって想定された特定状態により、命令復号論理回路８１０からの復号された指令と一緒に、書込みまたは読取り指令信号は、構成レジス

タ８０６に、（アドレス復号論理回路８１４によって復号されたような）命令の６ビット・アドレスによって識別された６４のレジスタの一つで読取りまたは書込みを実行させるべく指令発生器論理回路８１２によって生成される。

【０１９７】ＭＰ１８（図１）によって供給されたデータは、マルチプレクサ８１６を介してＤＲ８０４に書込まれる。ＯＬＡＰ２５８を用いている論理回路は、二つの個別ソースからＤＲ８０４を書込みうるし、ＩＲ８０２に早めに書込まれた命令情報及び直列バス１９ＡのＴＣＫ及びＴＣＭ信号回線によって運ばれた信号(signalling)による状態マシン８００の動作を用いてＭＰ１８によってＤＲ８０４に選択的に結合されかつ書込まれたそれらのソースで３２ビット・レジスタを供給する。３２ビットＤＲ８０４は、適切な１１４９．１命令の使用と一緒に、“ＣＡＰＴＵＲＥ－ＤＲ”、“ＳＨＩＦＴ－ＤＲ”、及び“ＵＰＤＡＴＥ－ＤＲ”として１１４９．１に記載された状態を通して状態マシン８００をステップすることによってアクセスされる。命令の追加ビットは、ＤＲ８０４に、ＣＡＰＴＵＲＥ－ＤＲ状態によりチップ状態情報を含んでいるチップ内の選択された値を読取らせる。他の１１４９．１命令は、構成及び初期化目的のためにＵＰＤＡＴＥ－ＤＲ状態に選択されたレジスタに対してレジスタ内容をコピーさせる。ＤＲ８０４の内容は、１１４９．１　ＳＨＩＦＴ－ＤＲ状態を用いて（直列バス１９Ａを介して）ＭＰ１８と変換される。ＯＬＡＰ２５８の構成及び動作の更なる情報に対して、ＩＥＥＥ１１４９．１標準（ＩＥＥＥ１１４９．１－１９９０、５月、１９９０年、ＳＨ１３１４４）が参照されうる。

【０１９８】非対象変数：“非対象変数”は、他のものから一対のＣＰＵｓ１２の一つにおいて異なるか、または異なりうる、値である。非対象変数の例は、他のＣＰＵのものから異なる、ＣＰＵ一読取り可能位置、例えば、レジスタ外側メモリ２８か、または訂正可能メモリまたはキャッシュ誤りの発生をトラックするために用いられるレジスタの内容、に割り当てられかつ保持される通し番号を含むことができる（誤りを検出し、訂正し及び報告することは、デュプレックされたＣＰＵｓにロックーステップ同期を失わさせないものと想定する）。デュプレックス・モードでは、非対象変数の注意深い処理は、論理的の同等であると仮定して、（各ＣＰＵ１２のメモリ２８に維持された）システム・メモリの多重コピーが、常に同一のデータを含むことを確実にするために重要である。非対象変数が二つのデュプレックスされたＣＰＵｓ１２のそれぞれによって単に読取られたならば、次いでメモリへ書込まれ、各ＣＰＵのメモリ２８の内容は、それにより少なくともそれぞれによって読取られた値で異なる。一対のＣＰＵｓ１２を許容するために、デュプレックス・モードで動作し、非対象変数を処理すべく、“ソフトーヴォート(soft-vote)”（ＳＶ）論理回路素子９００（図４３）が各ＣＰＵ１２の各インターフェイス装置２４に供給される。図４３～図４４が示すように、各インターフェイス装置２４のＳＶ論理回路素子９００は、バス回線９０２ａ及び９０２ｂを含んでいる、２ビットＳＶバス９０２によって互いに接続される。バス回線９０２ａは、ＣＰＵ１２Ａのインターフェイス装置２４からＣＰＵ１２Ｂのそれらに１ビット値を運ぶ。逆に、バス回線９０２ｂは、ＣＰＵ１２ＢのＳＶ論理回路素子９００からＣＰＵ１２Ａのそれらに１ビット値を運ぶ。

【０１９９】図４４に示すのは、ＣＰＵ１２Ａのインターフェイス装置２４ａのＳＶ論理回路素子９００ａである。各ＳＶ論理回路素子９００は、示さない限り、論理回路素子９００ａの説明が（ＣＰＵ１２Ａ，インターフェイス装置２４ｂの）他の論理回路素子９００ａ、及び（ＣＰＵ１２Ｂの、インターフェイス装置２４ａ，２４ｂの）９００ｂにも同様に適用されるということが理解されるべきであるように各他のＳＶ論理回路素子９００に対して構成及び機能において実質的に同一である。図４４が示すように、ＳＶ論理回路素子９００ａは、４つの１ビット・レジスタを含む：出力レジスタ９０４、局所入力レジスタ９０６、遠隔入力レジスタ９０７、及び出力イネーブル・レジスタ９１２。出力レジスタ９０４は、共有バス回線９０２ａに、マルチプレクサ（ＭＵＸ）９１４及び３状態ドライバ９１８を介して、結合される。ＣＰＵ１２Ａの論理回路素子９００ａだけが、バス回線９０２ａを駆動し、次いで二つの論理回路素子に一つだけがバス回線を駆動する。どれか一つは、イネーブル・レジスタ９１２の内容に依存する。従って、バス回線９０２ａは、論理回路素子９００ａの出力レジスタ９０４を、ＣＰＵ１２Ｂの論理回路素子９００ｂのそれぞれの遠隔入力レジスタ９０７に伝達する。バス回線９０２ａは、論理回路素子９００ａの一つの（マルチプレクサ９１４及びドライバ９１６を介して）出力レジスタ９０４を、論理回路素子９００ａの他の（並びにそれ自身の）局所入力レジスタに伝達する。この方法で、ＣＰＵ１２Ａの二つのインターフェイス装置２４ａ，２４ｂは、互いに非対象変数を伝達することができる。

【０２００】同様なファッションで、ＣＰＵ１２Ｂの論理回路素子９００ｂの出力レジスタ９０４は、論理回路素子９０２ａ（及び他のインターフェイス装置２４ｂのもの）の遠隔レジスタ９０７にバス回線９０２ｂによって伝達される。論理回路素子９０２は、一対の構成レジスタ７４（図９）を形成する。それゆえに、それらは、出力レジスタ９０４及び／又はイネーブル・レジスタ９１２を選択しかつ書込むためか、または入力局所及び遠隔レジスタ９０６及び９０７を選択しかつ読取るために少なくともアドレス／データ・バス７４（図４４にバス７４’として示す）の一部にわたる必要なデータ／アド

レス情報を伝達することによってプロセッサ装置２０によって書込まれうる。ＭＵＸ９１４は、ＳＶ論理回路素子９００ａに対するバス回線９０２ａの選択的使用をＣＰＵ１２Ａの各インターフェイス装置２４に供給すべくか、または一対のＣＰＵｓ１２をロックーステップ、デュプレックス動作に至らせるために用いれる（以下に説明する）再統一処理の間中に遭遇したならばＢＵＳ　ＥＲＲＯＲ信号を伝達するために動作する。出力イネーブル・レジスタは、３状態ドライバをイネーブルする（またはディスエーブルする）ビットで書込まれ、それがＳＶ出力レジスタ９０４の内容でバス回線９０２ａを駆動する。

【０２０１】上述したように、ＳＶ論理回路素子９００は、デュプレックス・モードで動作しているときにＣＰＵｓ１２Ａ，１２Ｂに、非対象変数のビット毎の交換を実施される。ＣＰＵｓ１２Ａ，１２Ｂがデュプレックス・モードであるときには、それらは、両方とも、もし同じ瞬間でなければ、実質的に同じ仮想瞬時(virtual moment in time)に同一命令ストリームの同じ命令を実行していることを思い出す。それらの間の非対象変数の交換は、次の通りである。両方のＣＰＵｓは、命令ストリームに応じて、かつ本質的に同時に、各ＣＰＵの両方のインターフェイス装置２４の論理回路素子９００のイネーブル・レジスタ９１２を書込む。各ＣＰＵの二つの論理回路素子９００の一つは、関連ドライバ９１６をイネーブルする状態で書込まれる；他は、高いインピーダンス状態にドライバの出力を置く状態で書込まれる。それは、関連ドライバ９１６をイネーブルすべく書込まれる両方のＣＰＵｓ１２Ａ，１２Ｂのインターフェイス装置２４ａの論理回路素子９００と関連した出力イネーブル・レジスタ９１２であるということを想定する。それゆえに、各ＣＰＵのインターフェイス装置２４ａの出力レジスタ９０４は、バス回線９０２に伝達される；即ち、インターフェイス２４ａ（ＣＰＵ１２Ａ）の論理回路素子９００ａに関連した出力レジスタ９０４は、バス回線９０２ａに伝達され、ＣＰＵ１２Ｂのインターフェイス装置２４ａの論理回路素子９００ｂに関連した出力レジスタは、バス回線９０２ｂに伝達される。ＣＰＵｓ１２は、両方とも、それらの対応出力レジスタ９０４に非対象変数のビットを書込み、それぞれの関連遠隔入力レジスタ９０７の最大クロック・スキューを許容した後、読取りが後に続く。出力レジスタ９０４は、各ＣＰＵによって再び書込まれ、遠隔入力レジスタ９０７を読取ることが再び後に続く。この処理は、変数全体が各ＣＰＵ１２の出力レジスタ９０４から他の遠隔入力レジスタに伝達されるまで、一度に１ビット、反復される。ＣＰＵ１２Ｂの両方のインターフェイス装置２４は、非対象情報のビットを受信することに注目する。

【０２０２】ソフトーヴォート機構の使用の一例は、通し番号(serial numbers)の交換である。構成レジスタ７４の一つは、互いにデュプレックスされうる二つのＣＰＵｓのそれぞれを識別し、かつ互いにそれらを区別すべく始動（スタート・アップ）でセットされる１ビット・レジスタ（図示省略）である。それゆえに、一つのＣＰＵの１ビット・レジスタは、他のＣＰＵのものから異なる状態にセットされる。これは、そのＣＰＵに割り当てられた通し番号で装填される他の構成レジスタで、スタート・アップの間中も、続けられうる。通し番号に対して構成レジスタのどれが装填されたかは、１ビット識別レジスタの状態による。それゆえに、二つのＣＰＵｓは、それを一つのＣＰＵにおいてそれ自身の通し番号を有する“Ｒ１”（図示省略）と呼ぶその一つのレジスタを除き、それぞれそれらの通し番号を含んでいる二つの同一レジスタを有し、他のＣＰＵは、構成レジスタ“Ｒ２”（図示省略）にそれ自身の通し番号を有する。これらの値あデュプレックスされたＣＰＵｓによってメモリに書込まれることができる前に、Ｒ１，Ｒ２構成レジスタは、ソフトーヴォート機構を用いて、“調和”されなければならない。ＳＶ論理回路素子９００は、説明される再統一処理の間に発生しうるバス誤りを伝達するためにも用いられる。再統一が実行されるときには、ＲＥＩＮＴ信号がアサートされる。図４４に示すように、ＲＥＩＮＴは、ＭＵＸ９１４の制御（Ｃ）入力に印加される。それゆえに、ＲＥＩＮＴがアサートされるときには、ＢＵＳ　ＥＲＲＯＲ信号がＭＵＸ９１４によって選択されかつバス回線９０２ａに伝達される。

【０２０３】同期：個別に動作している（シンプレックス・モード）か、またはペアになりかつ同期ロックーステップ（デュプレックス・モード）で動作しているサブープロセッサ・システム１０Ａ，１０Ｂの適切な動作（図１及び図４）は、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，14B の間で伝達されるデータが適切に受信され、かつ（ＣＰＵｓ１２Ａ，１２Ｂの；図９）クロック同期ＦＩＦＯｓ１０２及び（ルータ１４Ａ，１４Ｂの；図２９）クロック同期ＦＩＦＯｓ５１９がデータまたは指令として誤って解釈されないとういことの確保が必要である。種々のクロック同期ＦＩＦＯｓ（ＣＰＵｓ１２の）１０２及び（ルータ１４の）５１８のプッシュ及びプル・ポインタは、少なくとも近周波数動作に対して初期化されることが必要である。一般に、これは、パワーが最初に印加され、プッシュ及びプル・ポインタ・カウンタをある規準距離離してセットし、ある既知の状態に関連ＦＩＦＯキューをプリセットするときに、パワー・オン信号（図示省略）によって通常のファッションで行われる。これがなされて、全てのクロック同期ＦＩＦＯｓが近周波数動作に対して初期化される。それゆえに、システム１０が最初にオンラインにされた（パワー・アップされた）ときに、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，１４Ｂとの間の通信リンクの動作は、近周波数モードである。

【０２０４】しかしながら、ＣＰＵｓ１２Ａ，１２Ｂがデュプレックス・モード動作にスイッチされるときに、更なるものが要求される。まず、各ＴＮｅｔリンク上のＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，１４Ｂとの間にデータを送付するために用いられるクロッキングは、周波数封じ込み動作にスイッチされなければならない。次に、デュプレックス・モード動作のロックーステップ動作を適切に実施するために、クロック同期ＦＩＦＯｓは、別の経路に見出されない一つの経路における遅延を収容するためにそれからそれらがデータを受信する特定ソースで動作すべく同期されなければならない。例えば、デュプレックス・モード動作は、ペアになったＣＰＵｓ１２が同じ仮想時間に同一命令ストリームの各命令を実行することを必要とするということを思い出す。（“仮想”時間とは、ペアになったＣＰＵｓ１２による同一命令の実際の実時間実行が、少しの量だけ異なっても、外側世界によって見られるときにそれらのアクションは、まったく同じであるということを意味する。）ルータ１４Ａ及び１４Ｂからの入力データは、ロックーステップ動作のコンテキストにおいて、ほとんど同時に二つのＣＰＵｓによって受信されなければならない。ルータ１４Ａ，１４Ｂの一つまたは別のものからＣＰＵｓ１２Ａ，１２Ｂへの通信経路における遅延は、考慮されなければならない。それは、メッセージ・パケット記号を受信し、通信経路において課せられうる遅延に対して調整し、デュプレックス・モード動作に必要なロックーステップ同期を維持すべく同期マナーで二つのＣＰＵｓに記号を与えるべく動作するペアになったＣＰＵｓ１２のクロック同期ＦＩＦＯｓ１０２である。

【０２０５】同様なファッションで、ＣＰＵｓ１２の一つからルータ１４Ａ，１４Ｂによって受信された各記号は、（以下に、更に説明する）ＣＰＵｓの可能な発散を検査するために他からのものと比較されなければならない。それは、二つのＣＰＵｓ１２から受信した記号がクロック同期ＦＩＦＯｓから同時に検索されるように通信経路における遅延を収容すべく調整するＣＰＵｓ１２からメッセージ・パケットを受信するルータ１４Ａ，１４Ｂのクロック同期ＦＩＦＯｓ５１８の機能である。ＣＰＵｓ及びルータのクロック同期ＦＩＦＯｓがどのようにリセットされ、初期化され、かつ同期されるかを説明するまえに、同期ロックーステップ・デュプレックス・モード動作を維持するためのそれらの動作の理解は、有用であると信ずる。それゆえに、図３５をちょっと参照すると、例えば、ルータ１４Ａからデータを受信するＣＰＵｓ１２Ａ，１２Ｂのクロック同期ＦＩＦＯｓ１０２が示されている。図３５は、次いで、ルータ１４Ａからのデータ／指令記号及びクロックを二つのデュプレックスされたＣＰＵｓ１２Ａ，１２Ｂにそれぞれ結合する、１０ビット・バス32_x及び32_yに接続されたルータ１４Ａのポート出力5044及び5045を示す。メッセージ・パケットがＣＰＵ１２を識別している単一宛先アドレスを有しうるけれども、パケットは、記号毎に、ルータ１４Ａによって複製され、かつ両方のＣＰＵｓ１２Ａ及び１２Ｂの実質的に同時に送信されるということを思い出す。

【０２０６】二つのＣＰＵｓ１２Ａ，１２Ｂは、ルータ１４ＡからＣＰＵｓの一つ（例えば、ＣＰＵ１２Ｂ）によって受信された記号が、他のＣＰＵ（ＣＰＵ１２Ａ）による（ルータによって複製されたような）同一記号のそれらの受信に関して未知（しかし最大）量の遅延を経験するように配置されるであろう。この遅延は、ルータ１４ＡからＣＰＵ１２Ｂへ、記号及び付随送信機クロックＴ＿Ｃｌｋを伝達するバス32_yの６４０で表される。デュプレックス動作の間中に同一記号ストリームを受信するためのクロック同期ＦＩＦＯｓ102_x，102_yの動作を考える。以下の、表６は、その動作を示す。簡単のために、表６は、遅延６４０が送信クロック（Ｔ＿Ｃｌｋ）の二期間以上でないということを想定する。しかしながら、遅延６４０がＴ＿Ｃｌｋの二クロック時間以上であるならば、キュー１２６の深さは、プッシュ及びプル・ポインタ・カウンタ１２８及び１３０の内容の間で増加した距離を供給すべくそれゆえに増加されなければならない。例えば、遅延６４０が記号のＣＰＵ１２Ｂでの到着が、ＣＰＵ１２Ａでの同じ記号の到着よりも３Ｔ＿Ｃｌｋ期間だけ大きいようであるならば、プッシュ及びプル・ポインタ・カウンタ間の距離は、少なくとも４であるべきである。それゆえに、そのような場合における、キュー１２６の深さは、６記号位置、またはそれ以上である。

【０２０７】

【表６】

表　６

項　目	RST	clk 1	clk 2	clk 3	clk 4	clk 5	clk 6	clk 7
CPU 12A 値								
プッシュポインタ	0	1	2	3	0	1	2	3
プルポインタ	2	3	0	1	2	3	0	1
バイト 0	IDLE	A	A	A	A	E	E	E
バイト 1	IDLE	IDLE	B	B	B	B	F	F

バイト 2	IDLE	IDLE	IDLE	C	C	C	C	G
バイト 3	IDLE	IDLE	IDLE	IDLE	D	D	D	D
アウト reg	IDLE	IDLE	IDLE	A	B	C	D	E
CPU 12B 値								
プッシュポインタ	0	0	1	2	3	0	1	2
プルポインタ	2	3	0	1	2	3	0	1
バイト 0	IDLE	IDLE	A	A	A	A	E	E
バイト 1	IDLE	IDLE	IDLE	B	B	B	B	F
バイト 2	IDLE	IDLE	IDLE	IDLE	C	C	C	C
バイト 3	IDLE	IDLE	IDLE	IDLE	IDLE	D	D	D
アウト reg	IDLE	IDLE	IDLE	A	B	C	D	E

【０２０８】表６の上半分の行は、ＣＰＵ１２Ａ（インターフェイス装置２４ａ）に対するプッシュ及びプル・ポインタ・カウンタ１２８，１３０によって保持される値、キュー１２６の４つの記憶位置（バイト０，．．．，バイト３）の各々の内容、初期リセット（ＲＳＴ）期間及び送信機クロック、Ｔ＿Ｃｌｋの後続クロック・サイクルに対する出力レジスタ１３２の内容を示す。表６の下半分の行は、複製された記号ストリームの各記号に対するＣＰＵ１２Ｂインターフェイス装置２４ａのＦＩＦＯ１０２$_y$に対して同じものを示す。遅延６４０が２Ｔ＿Ｃｌｋ期間以上でないと想定して、（カウンタ１２８，１３０で維持される）プッシュ及びプル・ポインタは、二位置(two locations) 離れたキュー１２６の位置を指し示す。プッシュ・ポインタ・カウンタ１２８は、それぞれ受信した記号が記憶されるキュー１２６の次の位置を指し示し、かつプル・ポインタ・カウンタ１３０は、それぞれ記号がキューからプルされる位置を指し示す。表６、図３５を参照して、“ＩＤＬＥ”記号のストリームをそれが先に送っているプロトコルに付いている、ルータ１４Ａは、記号Ａで始まる、記号ストリーム（メッセージ・パケット）を送り始める、ということをここで想定する。表６が示すように、記号Ａは、遅延６４０によりＣＰＵ１２Ａでのその到着よりも一サイクル後でＣＰＵ１２Ｂに到着する。しかし、ＣＰＵ１２Ｂに対するプッシュ・ポインタ・カウンタ１２８の内容がＣＰＵ１２Ａのものに従い、それもまた一サイクル遅延する、ということに注目する。それゆえに、ＣＰＵ１２Ａでのその到着よりも一サイクル後で記号ＡがＣＰＵ１２Ｂに到着しても、両方は、キュー１２６の“バイト０”位置に記憶される。これは、（１）ＦＩＦＯｓ１０２が（以下に説明する処理を）同期で動作すべく同期され、かつ（２）プッシュ・ポインタ・カウンタ１２８が記号のソース、即ち、ルータ１４ＡからのＴ＿Ｃｌｋによって生成されたクロック信号によってクロックされ、そのクロック信号が記号によって経験されたものと同じ遅延６４０に出会うからである。他方、プル・ポインタ・カウンタ１３０は、それらがＣＰＵｓ１２のパケット受信機９４によって生成された局所受信機クロック（Ｒｃｖ　Ｃｌｋ）によってクロックされるので、互いに常にマッチする。更に、これらの局所受信機クロックは、動作のデュプレックス・モードであるときに周波数及び位相封じ込みされる；それらは、いかなる遅延も経験しない。

【０２０９】遅延６４０を見る別の方法は、ルータ１４ＡとＣＰＵ１２Ｂの間の通信経路（バス３２$_y$）におけるパイプラインの一部としてそれを考えることである。遅延６４０は、最大遅延が記号にその記号がキューからプルされる少なくとも１クロック・サイクル前に記憶キュー１２６をエンター（入力）させる限り、あらゆる値のものでありうる。ＣＰＵ１２Ａに伝達された記号は、実質的に、その複製がＣＰＵ１２Ｂのキュー１２６からプルされると同時にキュー１２６からプルされる前に１エキストラ・サイクル待つ。それは、ルータ１４Ａによって送信された記号ストリームの各記号が同時にＣＰＵｓ１２Ａ，１２Ｂのクロック同期ＦＩＦＯｓ１０２からプルされ、デュプレックス・モードで動作しているときに受信したデータの要求された同期を維持するというこの方法においてである。実質的に、ＣＳ　ＦＩＦＯｓ１０２のキュー１２６の深さは、ルータ１４ＡからＣＰＵｓ１２Ａ，１２Ｂへの二つの経路に同じ遅延を与えるべく調整する。表６を参照して説明された動作を達成するために、図４５に示されたリセット及び同期処理が用いられる。処理は、デュプレックス・モード動作に対してＣＰＵｓ１２Ａ，１２Ｂのクロック同期ＦＩＦＯｓ１０２を初期化するだけでなく、デュプレックス動作に対してルータ１４Ａ，１４Ｂの各々のＣＰＵポートのクロック同期ＦＩＦＯｓ５１８（図２７）を調整すべく動作もする。リセット及び同期処理は、ＳＹＮＣ　ＣＬＫ信号９７０（図４６）によって表された、時期を始動し、ＣＰＵｓ１２Ａ及び１２Ｂ並びにルータ１４Ａ，$_{14B}$の対応クロック同期ＦＩＦＯｓをリセットしかつ初期化すべくＳＹＮＣ指令記号を用いる。（ＳＹＮＣ　ＣＬＫ信号は、システム１０の素子、特にルータ１４Ａ，１４Ｂ及びＣＰＵｓ１２Ａ，１２Ｂへの分配（分布）のためにクロック発生器６５４（図３６）によって生起される。それは、クロック同期ＦＩＦＯｓにより記号を受信すべく用いられる、Ｔ＿Ｃｌｋよりも低い周波数のものである。例えば、Ｔ＿Ｃｌｋがおおよそ５０Ｍｈｚであると

ころでは、ＳＹＮＣ　ＣＬＫ信号は、おおよそ３．１２５ＭＨｚである。）

ここで図４５を参照すると、リセット及び初期化処理は、それらが同じクロック信号から導出されるようにＣＰＵｓ１２Ａ，１２Ｂ及びルータ１４Ａ，１４Ｂによって用いられたクロック信号を送信（Ｔ＿Ｃｌｋ）及び装置の局所クロック（Ｌｏｃａｌ　Ｃｌｋ）クロック信号としてスイッチすることによってステップ９５０で開始する。Ｔ＿Ｃｌｋ及びＬｏｃａｌ　Ｃｌｋ信号は、実質的に同じ周波数であるが、種々のクロック信号を伝達することにおいて固有な遅延により同相である必要はない。更に、ＣＰＵｓ１２Ａ，１２Ｂにおける構成レジスタ（インターフェイス装置２４の構成レジスタ７４）及びルータ１４Ａ，１４Ｂにおける（ルータ１４Ａ，１４Ｂの制御論理回路装置５０９に含まれた）構成レジスタは、ＦｒｅｑＬｏｃｋ状態にセットされる。

【０２１０】以下の説明は、ステップ９５２を含み、かつインターフェイス装置２４（図９）、ルータ１４Ａ（図２７）、図４５及び図４６を参照する。周波数封じ込み動作におけるクロックで、ＣＰＵ１２Ａは、ＳＬＥＥＰ指令記号を送り始めることをそれに指令すべくオフ・ラインＣＰＵ１２Ｂにメッセージ・パケットを送る。次に、また、ＣＰＵ１２Ａは、ルータ１４ＡにＳＬＥＥＰ指令記号を送り始め、さもなければ送られうるＲＥＡＤＹ指令記号を置換し、自己アドレス指定されたメッセージ・パケットが続く。ＳＬＥＥＰ指令記号が受信されかつルータ１４Ａによって認識されたときに受信されかつ再送信される処理のメッセージ・パケットは、終了させられる。しかしながら、更なるメッセージ・パケットは、一つを除いて、近づけられない（離される(held off)）：ＣＰＵ１２Ａからの自己アドレス指定されたメッセージ・パケット。それらのメッセージ・パケットは、受信され、かつ（宛先アドレス毎に）ＣＰＵ１２Ａへルータ１４Ａによって戻される。ＳＬＥＥＰ指令記号は、同期処理に対してルータ１４Ａを“静止(quiece)”すべく動作する。ＣＰＵ１２Ａによって送られた自己アドレス指定されたメッセージ・パケットは、ＣＰＵ１２Ａによって受け戻されたときに、ＳＬＥＥＰ指令記号の後に送られた自己アドレス指定されたメッセージ・パケットがルータ１４Ａによって最後に処理されなければならないので、ルータ１４Ａが静止状態であることをＣＰＵに知らせる。

【０２１１】ステップ９５４では、ＣＰＵ１２Ａは、ＳＬＥＥＰ指令記号を送ることの始動に続いてそれが送った自己アドレス指定されたメッセージ・パケットをそれが受け戻したかどうかを調べる。それがそのメッセージ・パケットの戻りを見て、かつそれによりルータ１４Ａが更なるメッセージ・パケットを一時的に処理しないことを確信したときに、ＣＰＵ１２Ａは、ＳＹＮＣ指令記号をルータ１４Ａに送るべくステップ９５６をエンター（入力）する。そのＳＹＮＣ指令記号がルータによって受信され、かつ指令復号論理回路５４４（図２９）のようなものによって認識されたときに、制御論理回路５０９は、知らされる。制御論理回路５０９は、ＣＰＵｓ１２Ａ，１２ＢにエコーバックされるＳＹＮＣ指令記号を生成すべく（ステップ９５８）、ポート出力5044，5045の指令記号発生器６３２（図３４）に信号を送るために、ＳＹＮＣ　ＣＬＫ９７０の次の立上りエッジ（時間t_1－　図４６）を待つ。次に、ステップ９６０（及びＳＹＮＣ　ＣＬＫ９７０の時間t_2）で、ルータの制御論理回路５０９は、ＣＰＵｓ１２Ａ，１２Ｂから直接的に記号に受信するルータの入力論理回路5054，5055に含まれる二つのクロック同期ＦＩＦＯｓ５１８に印加されるＲＥＳＥＴ信号９７２をアサートする。ＲＥＳＥＴは、アサートされる間、パワー・オン・リセット処理に関連して上述したように、互いに離れた所定数（この例では、２）離れた位置を記憶キュー５１８の位置で指し示すべく既知の状態にセットされたプッシュ及びプル・ポインタ・カウンタ５３０，５３２（図２９）で一時的非動作リセット状態に二つのクロック同期ＦＩＦＯｓ５１８を保持する。

【０２１２】同様に、ＳＹＮＣ記号は、ルータ１４Ａ，１４ＢによりＣＰＵｓ１２にエコーバックされる。ＣＰＵｓ１２受信ＳＹＮＣ記号のそれぞれがパケット受信機９６の記憶及び処理装置（図９及び図１０）によって検出されると、ＲＥＳＥＴ信号を各ＣＰＵ１２のパケット受信機９６（実際には、記憶及び処理素子１１０；図１０）によりアサートさせる。ＲＥＳＥＴ信号は、ＣＰＵｓ１２のクロック同期ＦＩＦＯｓ１０２（図１０）に印加される。このＣＰＵ　ＲＥＳＥＴ信号は、リセット状態に両方のＣＰＵｓ１２のＣＰＵクロック同期ＦＩＦＯｓ１０２を同様に保持し、それらの記憶キュー１２６（図１１）、及びプッシュ及びプル・カウンタ１２８，１３０を既知の状態に置く。ステップ９６２では、ＳＹＮＣ　ＣＬＫ９７０信号の時間t_3で、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，１４Ｂとの間の記号送信を伴う送信機クロック信号（Ｔ＿Ｃｌｋ）は、一時的に停止される。ステップ９６３（時間t_4）で、ＣＰＵｓ１２及びルータ１４Ａ，１４Ｂは、ＲＥＳＥＴ信号をディアサートし、かつＣＰＵｓ１２Ａ，１２Ｂ及びルータ１４Ａ，１４Ｂのクロック同期ＦＩＦＯｓは、それらのリセット状態から解放される。ステップ９６４（t_5）で、ルータ１４Ａ及びＣＰＵ１２は、Ｔ＿Ｃｌｋへの送信を再開しかつリンク上の最大予想遅延(maximum expected delay)に対する調整を許容するショート構成可能遅延(short configurable delay)を始める。遅延の終りで、ルータ１４Ａ及びＣＰＵｓ１２は、それらの対応クロック同期ＦＩＦＯｓからデータをプルすることを再開しかつ通常の動作を再開する。ルータ１４Ａのクロック同期ＦＩＦＯｓは、（ＩＤＬＥ記号にＲＥＳＥＴによっ

て予めセットされた）キューから記号をプルすることを開始し、かつＴ＿Ｃｌｋは、キューに記号をプッシュすることを開始する。Ｔ＿ＣｌｋでＣＰＵ１２Ａから受信した最初の記号は、例えば、付随Ｔ＿Ｃｌｋ信号でキュー位置０（またはプッシュ・ポインタ・カウンタがリセットされた値により指し示された他の位置）においてクロック同期ＦＩＦＯにプッシュされるということに注目する。同様に、ＣＰＵ１２Ｂからの最初の記号は、ＦＩＦＯキューの位置にそしてまた位置０（またはＲＥＳＥＴによりプッシュ・ポインタ・カウンタがセットされた他の位置）において配置される。ルータ１４Ａのクロック同期ＦＩＦＯｓは、ルータ１４Ａ及びＣＰＵｓ１２Ａ，１２Ｂの間で、他のものに関して、一つの通信経路に存在しうるいかなる遅延６４０を収容すべくここで同期される。

【０２１３】同様に、同じ仮想時間において、両方のＣＰＵｓ１２Ａ，１２Ｂのクロック同期ＦＩＦＯｓ１０２の動作は、再開され、それらをルータ１４Ａに同期する。また、ＣＰＵｓ１２Ａ，１２Ｂは、ＲＥＡＤＹ記号のためにＳＬＥＥＰ指令記号を送ることを中止して、適当なときに、メッセージ・パケット送信を再開する。それは、ルータ１４Ａに対する同期処理を終了する。しかしながら、処理は、ルータ１４Ｂに対しても実行されなければならない。それゆえに、ＣＰＵ１２Ａは、ステップ９５２に戻りかつ今回はルータ１４Ａの代わりにルータ１４Ｂでステップ９５２－９６６を再び実行し、その後全てのＣＰＵｓ１２Ａ，１２Ｂ及びルータ１４Ａ，１４Ｂは、周波数封じ込みモードで動作すべく初期化される。デュプレックス・モード動作に対して残っているものは、二つのＣＰＵｓ１２Ａ，１２Ｂを同じ動作状態に置き、かつ本質的に同じ瞬間にそれらに同じ命令を実行する。再統一と呼ばれる、二つのＣＰＵｓを同じ動作状態に置くことは、以下に説明する。しかしながら、まず、ＣＰＵｓ１２Ａ，１２Ｂがデュプレックス・モードで動作していると想定して、デュプレックス動作からＣＰＵｓの発散を結果として生ずる、可能な誤りを検出しかつ処理するために用いる方法及び装置を説明する。

【０２１４】発散検出及び処理：デュプレックス・モード動作は、ＣＰＵレベルでフェイル機能的フォルトトレラントを実施する。一対のデュプレックスされたＣＰＵｓ（例えば、システム１０のＣＰＵｓ１２Ａ，１２Ｂ　－　図１）の各々は、他の実質的に同一複製（コピー）であり、状態及びメモリ内容を含み、かつ両方ともに、実質的に同時に、同一命令ストリームの同一命令を実行し、論理的、フォルトトレラントＣＰＵを形成する。ＣＰＵｓ１２Ａ，１２Ｂの一つまたは他のものの故障(failure）は、そのフォルト（故障）が検出されかつ適切に処理される限り　－　システム１０の動作を停止、またはスローダウンすらしない。故障しているＣＰＵの検出は、デュプレックス・モード動作の明らかな結果（帰結）を用いる：両方のＣＰＵｓ１２Ａ，１２ＢのＩ／Ｏ出力は、適切なデュプレックス・モードに対して記号毎に同一である。それゆえに、適切な継続デュプレックス動作を確信するためになされることが必要な全てのことは、デュプレックスされたＣＰＵｓのＩ／Ｏ出力を記号毎に比較することである。故障しているＣＰＵは、他の動作の状態から発散し、結果的にその発散は、ＣＰＵｓのＩ／Ｏ出力にそれ自身を明らかにする。

【０２１５】図４７は、ルータ１４Ａ，１４Ｂで発散を最初に検出するために用いる手順を示し（ステップ１０００，１００２）、次いで、できるだけ早く故障しているＣＰＵを終了すべく優雅なマナーでその発散を処理し、かつシステム１０の残りに不良データを伝播することからそれを排除する。それゆえに、図４７のステップ１０００は、一つの論理ＣＰＵとしてロックーステップ同期で動作するＣＰＵｓ１２Ａ，１２Ｂ（図１）のデュプレックスされたペアを有する。周期的に、ＣＰＵｓ１２Ａは、サブープロセッサ・システム１０Ａ，１０Ｂの一つまたは他の周辺装置向けメッセージ・パケットを介してＩ／Ｏデータを送信する。出力メッセージ・パケットの宛先により、ステップ１００２は、ルータ１４Ａまたは１４Ｂの一つがそのＩ／Ｏデータを受信しているのを見て、それが受信されたならばＣＰＵ１２Ａからのメッセージ・パケットの各記号をＣＰＵ１２Ｂからのものと比較する。比較は、通常設計の比較回路（図示省略）によりＣＰＵｓ１２Ａ，１２ＢからＩ／Ｏを受信すべく接続されたポート入力５０２_4及び５０２_5の入力論理回路５０５の出力で行われる。受信した記号が同じであるならば、手順は、ステップ１０００及び１００２に残る　－　適切な動作を示す。

【０２１６】比較ステップ１００２が異なる記号を検出したならば、ルータ１４の比較回路（図示省略）は、ルータ制御論理回路５０９にＥＲＲＯＲ信号を発行し、発散を検出しているルータ１４が両方のＣＰＵｓ１２Ａ，１２ＢにＤＶＲＧ指令記号を送信するステップ１００４に処理を移動させる。好ましくは、ルータは、発散を知らせることとどのＣＰＵが継続するものであるかを知らせることとの間の時間を最小にすべくＤＶＲＧ記号を送る前にできるだけ長く待つ。脇道にちょっとそれて、発散を検出するこの技術によって達成される求められた多数の対照的ゴールを説明することは、この地点で利点的でありうる：まず、ルータ１４Ａまたは１４Ｂは、システムの残りへの誤りの伝播を防ぐために速やかな措置をとることが必要である。それゆえに、発散が検出されたにもかかわらず、ルータは、メッセージ・パケットの終了記号を除いて、その指定された経路にメッセージ・パケットを継続するかまたは送る：状態記号、“このパケットは、不良”（ＴＰＢ）または“このパケットは、良好”（ＴＰＧ）状態記号。この記号なしで、ダウンストリーム（下流）宛先は、受信したメッセージ・パケット

を用いない。

【０２１７】第２に、できるだけ少ないメッセージ・パケットが分断されなければならない。以下に更に説明するように、ＣＰＵｓ１２Ａ，１２Ｂの一つは、“気に入った”または主ＣＰＵと指定され、かつＣＰＵｓがデュプレックス・モードで動作しているときに、ルータの気に入ったＣＰＵからのメッセージ・トラフィックだけが送信される。分断は、もしあれば、どのＣＰＵがフォルトでありうるかの、決定を行うことができるまで、発散を検出するにもかかわらず、ルータにメッセージ・パケットを送信することを終了させることによって最小にされる。気に入ったＣＰＵでないならば、メッセージ・パケットは、終了記号の送信によって解放される　－　この場合ＴＰＧ記号。第３に、発散を検出しているルータは、どの誤りが発散を生成するためにトランスパイアされた(transpired)かを正確に決定することが必要である。それは、これに簡単なリンク誤り、リンク－レベル“キープアライブ”記号の損失、及びＣＲＣ誤りを探させる。ＣＰＵｓ１２は、それらが、発生すべきリンク－レベル・キープアライブ・タイム・アウトに対してＤＶＲＧ記号を受信した後に十分な時間を許容する。（簡単なリンク誤りも検出することなく）発散を検出しているルータは、ＤＶＲＧ記号で発散を知らせる前にメッセージ・パケットの終りを待つことによって受信したメッセージ・パケットのＣＲＣを検査する時間をそれ自身が買う。

【０２１８】最後に、かつ第４に、システム１０は、ＴＮｅｔトランザクション・タイムアウトまたは支持不可能Ｉ／Ｏ遅延をもたらすことを回避するためにショート・バウンディド(short bounded) 時期に発散処理を終了しなければならない。このゴールは、ＣＰＵからのメッセージ・パケットの結論を待つことがかなりの量の時間を費やすように、（終了状態記号の送信を見合わせることにより）メッセージ・パケットの解放の保持とある程度矛盾する。しかしながら、そのような遅延は、メッセージ・パケットを送信するためのＣＰＵに対する最悪ケース時間(worstcase time) が保証されるならば、ＴＮｅｔタイムアウトをもたらすことができない。ＣＰＵｓ１２は、ＤＶＲＧ記号の受信により、それぞれが、その内でＣＰＵｓ１２がそれらのいずれが故障したかを決定することを試み、かつ動作を終了し、かつそれらのどれが継続すべきかを決定することを試みる所定時期を制定するために用いられるタイマーをスタートする（ステップ１００６）。更に、両方のＣＰＵｓ１２Ａ，１２Ｂは、両方のルータ１４Ａ，１４ＢにＤＶＲＧ指令記号をエコーバックする。このエコーバックされたＤＶＲＧ記号を受信する、ルータ１４Ａ，１４Ｂが発散を検出しないか、またはＤＶＲＧ記号を予め見ていないならば、それは、ＤＶＲＧ記号をＣＰＵｓにもエコーバックする。この方法でＤＶＲＧ指令記号をエコーバックすることは、ＣＰＵｓ１２及びルータ１４Ａ，１４Ｂが全てＤＶＲＧ記号を見ておりかつ可能な発散に気付いていることを確実にする。

【０２１９】一度、発散が検出され、ＣＰＵｓ（またはルータ）の一つの故障が示されることが全ての考慮したもの（ＣＰＵｓ１２Ａ，１２Ｂ及びルータ１４Ａ，１４Ｂ）に対して明瞭であるならば、その故障のあらゆる結果が、不良データの形で、システム１０の残りに伝播されないということを確実にするために配慮が払われなければならない。同時に、システム１０は、フォルトのトレラントでありかつ走行し続けなければならない。それゆえに、（ＣＰＵｓからの）出力パケット送信は、ルータが発散を検出と同時にＣＰＵｓからくるメッセージ・パケットが良好か不良かを決定することができるまで、少なくとも部分的に、続けなければならない。更に、発散をもたらした(divergence-causing)ＣＰＵは、決定され、かつ透過的に（即ち、外部干渉なしに）システムから除去されなければならない。この後者のタスクは、ＣＰＵｓ１２の責任であり、一度発散及びあらゆる誤りがＣＰＵｓ１２に知らされたならば、それらは、それらのどれが動作を続け、どれが更なる動作を終了しかつそれによってシステム１０からそれ自身を効果的に除去するかそれら自身の間で決定しなければならない。

【０２２０】それゆえに、発散ルーチンのステップ１００６は、ＣＰＵｓ１２Ａ，１２Ｂの各々にそれらに供給された種々の誤り表示を解析させる；この誤り解析は、以下に更に説明する。しかしながら、ちょっとの間、不良データの伝播を制限すべく発散を検出したルータ１４の機能は、説明を必要とする。ＤＶＲＧ記号がルータ１４から発行されたか、または受信された後、ＣＰＵｓから受信しかつ発散が検出されたか、またはＤＶＲＧ記号が受信されたときに送られる処理で受信した全ての更なるメッセージ・パケットは、パケットを終了する状態記号を除き、ルータを通過される；即ち、ＴＰＧ（このパケットは、良好）またはＴＰＢ（このパケットは、不良）状態インジケータ記号。デュプレックス動作の間中、上記で簡単に説明したように、ルータ１４Ａ，１４Ｂの各々は、制御論理回路５０９（図２７）に含まれる構成レジスタ（図示省略）にセットされたビット位置の側に“気に入った”ＣＰＵを有すべく構成される。デュプレックス動作では、ルータは、の気に入ったＣＰＵから受信したメッセージ・パケットを再送信する；他のまたは“気に入らない”ＣＰＵからのメッセージ・パケットは、発散検出に対してだけ用いられる。ルータは、ＴＰＧ／ＴＰＢ状態インジケータ記号を後ろに追加することによりパケットを“解放する”前に（ステップ１０１４）、それらのどれが動作を継続するか、どのルータ１４Ａ，１４Ｂが知らされるか（ステップ１０１２）というＣＰＵｓによってなされた決定を待たなければならない。ルータが気に入ったＣＰＵ１２が継続すべく決定さ

れたものであることを知らされたときに、ルータは、ＴＰＧ状態インジケータ記号を追加しかつ送ることによってメッセージ・パケットを解放する。逆に、ルータが別を、即ち、それが継続する気に入ったＣＰＵでないことを、知らされたならば、メッセージ・パケットは、ＴＰＢ記号を追加することによって廃棄される。

【０２２１】損失するデータの量を制限するために（上記第２のゴール）、二つのルータは、ことなる好み(favorities)で構成される（例えば、ルータ１４Ａの好みがＣＰＵ１２Ａであり、ルータ１４Ｂの好みがＣＰＵ１２Ｂである）。続けて、一度、検出された発散がＣＰＵｓ１２Ａ，１２Ｂ及びルータ１４Ａ，１４Ｂにブロードキャスト（放送）されたならば（ステップ１００４）、ＣＰＵｓ１２Ａ，１２Ｂの各々は、発散のフォルトがどこにあるかをそれぞれ個別に決定しようとしてステップ１００６で状況（状態）を評価し始める。一度、ＣＰＵｓ１２Ａ，１２Ｂのどれが故障したかが決定されたならば（ステップ１００８）、そのＣＰＵは、それ自身の動作を終了し（ステップ１０１２）、シンプレックス・モードにもかかわらず、動作を継続すべく他のものを残す。ＣＰＵｓ１２Ａ，１２Ｂがそれらのどれがフォルトでありうるかを検出されたかまたは報告された誤りから決定することができない場合には、それらは、各ＣＰＵのインターフェイス装置２４の構成レジスタ７４（図９）の一つに含まれる“タイ・ブレーカー”ビットに頼る（ステップ１０１０）。ステップ１００６にちょっと戻ると、ＣＰＵｓ１２Ａ，１２Ｂの故障しているものがどれであるかの決定は、ＣＰＵｓ１２Ａ，１２Ｂとルータ１４Ａ，１４Ｂをリンクしている通信経路上にどの誤りが検出されたかに主に基づく。ルータ１４Ａ，１４Ｂが発散に気付いた後、それらの各々は、上述したように、通常の動作を継続する：発散を示している記号差が検出されたか、またはその後受信されたときにＣＰＵｓ１２Ａ，１２Ｂからルータ１４Ａ，１４Ｂによって受信した単一メッセージ・パケットは、終結状態インジケータ記号を除きルータを通過される。両方のルータ１４Ａ，１４Ｂは、例えば、検出されたＣＲＣ誤り、指令記号誤り、またはリンク誤りの他の型のような、可能リンク・プロトコル誤り及び／又は違反に対して特定のルータ１４をＣＰＵｓ１２に相互接続している通信経路（ＴＮｅｔリンクＬ）を監視し続ける。そのような誤り／違反がリンクＬ（例えば、リンクＬｘ　－　図１）上で検出されたならば、検出ルータ１４Ａ，１４Ｂは、Ｔｈｉｓ　Ｌｉｎｋ　Ｂａｄ（このリンクは不良）（ＴＬＢ）記号をそのリンク（Ｌｘ）上のＣＰＵに送り戻す。同時に、検出ルータは、Ｏｔｈｅｒ　Ｌｉｎｋ　Ｂａｄ（他のリンクは不良）（ＯＬＢ）記号を（他のリンク、Ｌｙ上の）他のＣＰＵ１２にも送り戻す。それにより、両方のＣＰＵｓ１２は、ルータ１４Ａ，１４Ｂの一つへの通信経路（リンク接続Ｌｘ）、またはそのリンクに関連した何かが故障でありかつ信頼できないという事実を気付かされる。それは、どのＣＰＵが動作を終了すべきかを決定することを試みるために、ステップ１００６で、ＣＰＵｓが用いる情報であり、シンプレックス・モードにもかかわらず、他のＣＰＵを機能し続けさせる。ＣＰＵｓは、この誤り情報を累積し、以下の表７に示したものに類似する表を生起する。

【０２２２】

【表７】

表　7

	局所ルータ	遠隔ルータ	局所ＣＰＵのアクション
(1)	誤りなし	誤りなし	主に継続；代替は終了
(2)	誤りなし	ＴＬＢ又は誤り	終了
(3)	誤りなし	ＯＬＢ	継続；他のＣＰＵは終了
(4)	ＴＬＢ又は誤り	あらゆる状態	終了
(5)	ＯＬＢ	あらゆる状態	継続；他のＣＰＵは終了

【０２２３】表７は、ＣＰＵによって検出されうるかまたはそれに報告されうるし、かつそれによりＣＰＵが発散を知らされたにもかかわらず何をするかを決定しうる５つの可能条件を示す。表７に用いたように、“局所”は、ＣＰＵとして同じサブープロセッサ・システム１０Ａ，１０Ｂに含まれたルータ１４Ａ，１４Ｂを称することを意味する。例えば、図１を参照すると、ルータ１４Ａは、ＣＰＵ１２Ａに対して“局所”であるが、ＣＰＵ１２Ｂに対して“遠隔”である。表７に用いたように、“Ｐｒｉｍａｒｙ（主）”は、上記したタイブレーカー・ビットを称する：各ＣＰＵのインターフェイス装置２４（図９）の構成レジスタ７４の一つに含まれるビット。第１の状態にセットされたときに、その特定ＣＰＵ１２は、それにより、主（１次：ｐｒｉｍａｒｙ）として識別され、逆に他のＣＰＵ１２は、それが“代替”であることを示すべく反対の状態にセットされたその同じ構成ビットを有さなければならない。これらのビットは、ＣＰＵｓの別のものがその決定をすることができないときに継続すべくＣＰＵｓの一つを選択するために発散処理の間中に用いられる。局所ルータが誤りなしを報告する全ての場合に、ＣＰＵｓは、決定を行うことを遅延することに注目する。これは、他のＣＰＵが誤りを検出しかつ自己検査しえたし、かつ局所ルータがキープアライブ記号の損失を実質的に検出し、かつＯＬＢ記号により局所ＣＰＵに誤りを報告するという可能性を許容する。

【０２２４】“Ａｎｙ　Ｓｔａｔｕｓ（あらゆる状態）”は、それをちょうどを称する：遠隔ルータからの報告のいかんを問わず（誤りの表示、または誤りの無表示）、局所ＣＰＵは、見出し“Ａｃｔｉｏｎ　ｏｆ　Ｌｏｃａｌ　ＣＰＵ（局所ＣＰＵのアクション”の下に示されたアクションを行う。“Ａｃｔｉｏｎ　ｏｆ　Ｌｏｃａｌ　ＣＰＵ”は、ＣＰＵｓ１２Ａ，１２Ｂの特定のものによって取られたアクションを表わし、表の行の一つに示されたようにその特定ＣＰＵによって見られた条件を与える。例えば、行４にセットされた条件がＣＰＵ１２Ａ（ルータ１４Ａが誤りを報告したか、またはＣＰＵ１２Ａが誤りを検出した）によって見られたならば、ＣＰＵ１２Ａは、それがデュプレックスされたペアの他のもの、ＣＰＵ１２Ｂ、に動作を継続させるために、動作を終了すべきという決定を行う。逆に、行４の条件は、他のＣＰＵ１２Ｂは、ＯＬＢ記号をその“遠隔”ルータから受信することを示し、ルータ１４ＡとＣＰＵ１２Ａとの間の通信経路が疑わられるという事実を知らせている。ＣＰＵ１２Ｂの視点から、これは、行３または５によって表された条件でなければならない。ルータ１４Ａ，１４Ｂの一つだけ（この場合には、ルータ１４Ａ）が誤りを検出したならば、行３表示がＣＰＵ１２Ｂに与えられる。両方のルータ１４Ａ，１４Ｂが誤りを検出した場合には、それぞれは、ＴＬＢをＣＰＵ１２Ａにかつ ＯＬＢをＣＰＵ１２Ｂに知らせる。ＣＰＵ１２Ｂは、ルータ１４ＢからＯＬＢを見て、これを行５条件にマッチし、ＩＯＹ記号をルータ１４Ｂに発行して、継続する。

【０２２５】表７の行４及び５は、ある問題を許容しうることに注目する。例えば、ルータ１４ＡがＴＬＢをＣＰＵ１２Ａに知らせかつルータ１４ＢがＴＬＢをＣＰＵ１２Ｂに知らせたならば、両方のＣＰＵは、それら自身を殺す（凍結する）。しかしながら、両方の局所ルータが誤りを生起することが起こりえない場合のような、変わった想定ではない、ほとんどの場合一つのフォルトだけが所与の時間に発生すると想定したならば、表７の条件は、受容可能である。さもなければ、ルータ１４とＣＰＵ１２との間の二つ以上のリンク上の多重誤りが発生するならば、システムは、生き残る必要がない。同様に、両方のルータがＯＬＢｓをそれらの局所ＣＰＵｓに知らせたならば、両方のＣＰＵｓは、取って代わることを試みるであろう。これは、クロック故障を表わす傾向がある。クロック回路は、そのような誤りを検出しかつ故障しているＣＰＵを凍結すべきである。従って、表７は、ＣＰＵｓ１２及びルータ１４Ａ，１４Ｂが検出できる誤り表示を表わす。一般に、ＣＰＵ１２がその局所ルータから誤り表示を受信したならば、それは、ペアの他のものを継続させるために動作を終了する。主／代替指定に頼ることは、いずれのＣＰＵも、（ＤＶＲＧ指令記号の受信により始動された）各ＣＰＵのタイマーの期間満了においていかなる種類（行１、表７）の誤り表示を受信しないときにだけ生じる。この場合には、各ＣＰＵの主構成ビットに頼ることによりタイ（tie）がブレークされる。主として識別されたものは、継続しかつ他のものが終了されたと想定する；それ自身の構成ビットにより代替と識別されたＣＰＵは、その動作を終了する。

【０２２６】それゆえに、ＣＰＵｓ１２は、どれが継続し、かつどれが継続しないかの決定を行い（ステップ１００８）、次いで、ステップ１００６，１０１０の一つにおいてなされた決定により一つのＣＰＵが終了するステップ１０１２に続く。終了するＣＰＵ１２は、自己検査及び凍結を誘導することによってそのようにする。継続しているＣＰＵは、ルータが継続シテイルＣＰＵだけを頼り、他のＣＰＵからの全ての送信を無視すべきであるということをそれらに報告すべくＩＯＹ記号（Ｉ　Ｏｗｎ　Ｙｏｕ）をルータ１４Ａ，１４Ｂに送る。それに応じて、制御及び状態装置５０９（図２７）内の状態マシン（図示省略）は、上述した“気に入った”ビットを変える。２～３の例は、発散の概念を理解することを容易にする。図１を再び参照して、ＣＰＵｓ１２Ａ，１２Ｂがデュプレックス動作モードで動作し、ＣＰＵ１２Ａが全ての後続Ｉ／Ｏ動作がＣＰＵ１２Ｂのものとは異なるようにフォルトを被ることを想定する。従って、次のＩ／Ｏ送信で、ルータ１４Ａ，１４Ｂの一つ（データが指向されるもの；または故障がＣＰＵ１２Ｂのものとは異なる宛先にＩ／Ｏを指向しているＣＰＵ１２Ａを有するならば両方）は、発散を検出する。しかしながら、それらは、上述したように、パケット全体が現在のメッセージ・パケットＣＲＣ検査が通るかどうかを決定すべく受信されるまで、または簡単な誤りに出会うまで、待ち、その時に各ルータが両方のリンクＬ上のＤＶＲＧ記号を送信する。両方のルータはプロトコル誤りを見る（理解する）と想定する。検出されたプロトコル誤りは、両方のＣＰＵｓ１２にＤＶＲＧ記号を送っているルータ１４Ａ，１４Ｂをすぐに結果として生じ、誤りが検出されたことによりリンクＬ上にＴｈｉｓ　Ｌｉｎｋ　Ｂａｄ（ＴＬＢ）記号を送り戻す、即ち、リンクＬｘ，Ｌｙは、ルータ１４Ａ，１４Ｂを、それぞれ、ＣＰＵ１２Ａに接続している。それらがＴＬＢ記号を送ると同時に、両方のルータ１４Ａ，１４Ｂは、Ｏｔｈｅｒ　Ｌｉｎｋ　Ｂａｄ（ＯＬＢ）記号をＣＰＵ１２Ｂに送る。ＣＰＵ１２Ａは、ＤＶＲＧ記号の受信により、その記号をルータ１４Ａ，１４Ｂにエコーバックし、その内部発散処理タイマーをスタートし、継続すべきか終了すべきかの決定を始める。その局所ルータ１４ＡからＴＬＢを受信して、ＣＰＵ１２Ａは、それがＣＰＵ１２Ｂの継続を許容するために終了しなければならないということをすぐに決定する（行４、表７）。

【０２２７】更に、このシナリオにおいて、ＣＰＵ１２Ｂは、両方のルータ１４Ａ，１４ＢからＯＬＢ記号を受

信しかつそれが継続すべきＣＰＵであることを報告するそれらから決定する。従って、ＩＯＹ記号を両方のルータ１４Ａ，１４Ｂに発行する。それに応じて、ルータ１４Ａ，１４Ｂは、ＣＰＵ１２Ｂによるパケット送信だけが実行され、かつＣＰＵ１２Ａからの送信が無視されるようにそれら自身を構成する。また、発散検出は、故障しているルータを検出しうる。例えば、ルータ１４Ａは、それに発散作用を実行させるような方法で故障し、二つのＣＰＵｓ１２Ａ，１２Ｂを発散されると想定する。良好なルータ、ルータ１４Ｂは、この発散を検出し、かつＣＰＵｓのそれぞれにＤＶＲＧ記号でそれを知らせる。各ＣＰＵは、両方のルータ１４Ａ，１４ＢにＤＶＲＧ記号をエコーバックする。特定のルータ１４Ａの故障により、それは、ＤＶＲＧ記号をＣＰＵｓにエコーバックしうるしまたはしえない。ＣＰＵ１２Ａは、それがそれを故障したルータ１４Ａに接続しているその局所リンクに誤りを有することを見出し、それゆえにそれは、それが自己検査しかつ終了しなければならないことを決定する。逆に、ルータ１４Ｂは、この終了を検出し、かつＴＬＢ記号をＣＰＵ１２Ａに、かつＯＬＢ記号をＣＰＵ１２Ｂに戻す。次いで、ＣＰＵ１２Ｂは、両方のルータにＩＯＹ指令記号を発行する。

【０２２８】上述したのは、ルータ１４Ａ，１４Ｂの一つまたは他のものまたは両方が発散を検出し、ＣＰＵｓにＤＶＲＧ記号を発行するが、ＣＰＵｓ１２Ａ，１２Ｂまたはルータ１４Ａ，１４Ｂのどちらも誤りを検出しないような“クリーン”発散であった。従って、“主”ＣＰＵは、初期化の間中に構成レジスタに予めセットされたように、それが継続しかつ両方のルータ１４Ａ，１４ＢにＩＯＹ記号を発行しなければならないことを決定する。同時に、“代替”ＣＰＵ１２Ｂは、自己検査し、かつ終了する。上述したものに加えて、発散をもたらすことができる誤りまたはフォルトの型は、以下のものを含む：

－　訂正不可能なメモリ誤り、誤りの可能な伝播を排除するためにＣＰＵに動作をすぐに凍結させる。ＣＰＵは、ルータ１４Ａ，１４Ｂに対して死んだように見え、それらにＴＬＢ記号を機能ＣＰＵへ、かつＯＬＢを他の（稼動している）ＣＰＵへ送らせる。稼動しているＣＰＵは、それが継続することを決定し、かつＩＯＹ記号を両方のルータ１４Ａ，１４Ｂへ送る。

－　報告された誤りなしでＣＰＵｓ１２を発散させるソフトウェア欠陥。これは、（プロセッサ２０上で走っている）ソフトウェアが、状態を変更すべく既知の発散データを用いるときにのみ発生することができる。例えば、各ＣＰＵ１２が異なる通し番号を有するものと想定する（例えば、アドレス空間の読取り専用またはプログラマブル読取り専用領域に保持される）。ＣＰＵ１２Ａの通し番号は、ＣＰＵ１２Ｂのものとは異なる。プロセッサが、（例えば、通し番号がある値の後にくるならばブランチングにより）実行される命令のシーケンス（順番）を変えるために、またはプロセッサ・レジスタに含まれる値を変更するために通し番号を用いるならば、ＣＰＵｓ１２の完全な“状態”は、異なる。そのような場合には、“非対称変数”機構（例えば、ソフト・ヴォート機構）が用いられる。これは、それらが、同一の決定を行いかつ同一のデータに基づき同一の動作を実行することができるように、二つのＣＰＵｓ１２に情報を交換させる。この例では、ＣＰＵｓは、両方がコピーを有するようにそれらの通し番号の値を交換する。次いで、ＣＰＵ１２Ａの通し番号がある値の後にきたならば、それらは、コード（符号）を実行し、このシーケンスを実行する、等。主ＣＰＵの所定の構成、及び最終選択は、単に一つのＣＰＵに、及びそれによりシステム１０に、ソフトウェア介入なしで、処理を継続させる。

【０２２９】－　ＣＰＵ１２のインターフェイス装置２４の出力における誤りは、誤りが発散として発生するリンクＬにより、ルータ１４Ａ，１４Ｂによって検出され、かつＴＬＢ／ＯＬＢ状態は、ＣＰＵｓに戻され、ＤＶＲＧ記号が後に続く。ＣＲＣ誤り、及び指令記号汚染は、同様に検出及び処理される。

－　ＣＰＵ１２に接続するルータ１４Ａ，１４Ｂの出力上の誤りは、影響を及ぼされたＣＰＵのインターフェイス装置２４によって検出される。ＣＰＵは、故障しているルータ１４にＴＬＢ記号を送り、図４５及び図４６の発散処理ルーチンを起動すべくそのルータによる発散として検出される。誤りを検出しているＣＰＵは、終了し、かつ継続しているＣＰＵは、両方のルータにＩＯＹ記号を送る。発散は、システム１０（図１）に、デュプレックス・モード動作で動作しているときに、可能な故障を検出させ、かつ外部介入なしで、かつシステム・ユーザに透過的に、ユーザ・オペレーションを停止させることなくシステムへの誤りの伝播を未然に防ぐかまたは制限すべくシステムから故障している装置（ＣＰＵｓ１２Ａまたは１２Ｂ、またはルータ１４Ａ，１４Ｂ）を除去させる。故障したのがＣＰＵ１２であったならば、ＣＰＵは、回線からはずされ、かつ検査済みの、機能するＣＰＵで置き換えることができる。デュプレックス動作において、その代替装置(replacement)をオン・ラインに戻すことが次のセクションの主題である：再統一。

【０２３０】再統一：

－　概要

このセクションは、“再統一(reintegration)”を通して二つのＣＰＵｓ１２をロックーステップ、デュプレックス・モード動作に至らせるために用いる手順を説明する。説明は、図１に示された処理システム１０の部分を形成しているＣＰＵｓ１2Ａ，１２Ｂ、ルータ１４Ａ，１４Ｂ、及び保守プロセッサ１８Ａ，１８Ｂを参照する。更に、説明は、それらが機能する方法であるので、単一装置としてプロセッサ２０ａ，２０ｂ、インターフ

ェイス装置２４ａ，２４ｂ、及びＣＰＵｓ１２Ａ，１２Ｂのメモリ・コントローラ２６ａ，２６ｂ（図４）を参照する。再統一は、最初にオンラインにもってこられたか、またはある時間シンプレックス・モードで動作した後、またはシステム１０の先のデュプレックス・モード動作が発散を結果として生じ、かつ故障している素子（例えば、ＣＰＵｓの一つ）が除去されかつ置換された後に、デュプレックス・モード動作に二つのＣＰＵｓを置くために用いられる。

【０２３１】再統一は、まだ動作している（即ち、オン・ライン状態の）ＣＰＵｓ１２の一つで始めなければならず、外部介入なしで、再統一がバックグラウンドで実行され、従ってユーザに対して実質的に透過であるので、かなり確実にユーザ・アプリケーションを実行する。他のＣＰＵ１２は、それがユーザ・コードを走らせないという意味で、オフ・ライン状態である；それは、それにその初期化及び再統一に要求される最低限のタスクを実行させるべく十分なコードを走らせる。この初期化は、それらが、事実上同時に同一命令ストリームの同じ命令を実行することができるようにデュプレックス・モード動作に対して事実上同じ状態に一対のＣＰＵｓ１２を配置することを含み、結果として同じアクションをとる。また、再統一は、発散検出を実施することができ、かつＣＰＵｓ１２向けのメッセージ・トラフィックが事実上同時にペアになったＣＰＵｓの両方に送付されるように、ルータ１４Ａ，１４Ｂがデュプレックス・モード動作に対して構成されることを結果として生ずる。図４８～図５１のフロー図によってある程度より詳細に概説された、一つのオン・ラインＣＰＵのシンプレックス・モード動作から二つのＣＰＵｓのデュプレックス・モード動作に変更するための処理における主要ステップは、一般に次の通りである：

１．遅延した（“シャドウ”）周波数封じ込み、デュプレックス・モード動作に、二つのＣＰＵｓ（一つがオン・ライン、他のものがオフ・ライン）及びそれらの接続ルータをセットアップしかつ同期し、別個の(distinct)命令ストリームを実行する；

２．オン・ラインＣＰＵのメモリをオフ・ラインにコピーし、実行されておらず、かつオフ・ラインＣＰＵに、コピーされることが必要でありうる、オン・ラインＣＰＵのメモリにおける変化を監視するトラッキング処理を維持する；

３．同じ命令ストリームからの遅延した（スレーブ）デュプレックス・モードを走らせるべくＣＰＵｓをセットアップしかつ同期する（ロックーステップ動作）；

４．オン・ラインＣＰＵからオフ・ラインＣＰＵに全ての残っているメモリ位置をコピーする（このステップは、全てのメモリが読取られるまでオフ・ライン・メモリの各位置を読取り、かつオフ・ラインＣＰＵのそれらと異なると疑われるそれらのメモリ位置だけをコピーする）；そして

５．二つのＣＰＵｓの完全ロックーステップ、デュプレックス動作を起動する。

【０２３２】－　セットアップ

ここで図４８を参照すると、再統一手順が入力される前に、ＣＰＵｓ１２Ａ，１２Ｂ及びそれらの最初の回線ルータ（即ち、ＣＰＵｓに直接接続するもの）１４Ａ，１４Ｂがセットアップされなければならない。これは、ＭＰ１８Ａの使用を含む。ステップ１０５０では、ＭＰ１８Ａは、ＣＰＵｓ１２Ａ及び１２ＢのインターフェイスⅠ装置２４における制御レジスタ７４の所定のレジスタ（図示省略）を、両方のＣＰＵｓが周波数封じ込みモードであるが、一つ（オフ・ラインＣＰＵ）は、遅延したまたは“シャドウ”ファッションで動作し、他のものに遅れて多数（例えば、８）のクロック・サイクルを動作するような（ソフトウェア動作の後）次の状態に書込む。ＣＰＵｓ及びルータのこの動作のモードは、以後“シャドウ・モード(shadow mode)”と呼ばれる。ルータの構成レジスタ（図示省略）も、ステップ１０５２でＭＰ１８Ａによって同様にセットされる。更に、構成レジスタは、“気に入った”としてルータ１４Ａ，１４Ｂへオフ・ラインＣＰＵ１２Ａを識別するためにＭＰ１８Ａによって書込まれる。これは、ルータ１４Ａ，１４Ｂに、シャドウ・モードにおけるときに送信に対してＣＰＵ１２Ａだけを頼らせて、オフ・ライン１２Ｂから出うる全ての送信を無視する。

【０２３３】次に、上述したのと同じようなファッションでＣＰＵｓ１２Ａ，１２ＢのクロックⅠ同期ＦＩＦＯｓとルータ１４Ａ，１４Ｂを同期するシーケンスが入力され（ステップ１０６０－１０７０）、次いでそれらをシャドウ・モード動作に移動する。シャドウ・モード動作は、オフ・ラインＣＰＵ１２Ｂに送られた送信が記号毎にオン・ラインＣＰＵ１２Ａに送られたものに多数（例えば、８）のＴ＿Ｃｌｋクロック遅れるということを除き、二つのＣＰＵｓ１２Ａ，１２Ｂが真性デュプレックス・モード動作で機能しているときと同じようなファッションで、ルータ１４Ａ，１４Ｂから同じメッセージ・パケット及び他のＴＮｅｔ記号を受信する　ようなものである。即ち、ルータ１４Ａ，１４Ｂの一つから送信される記号は、その同じ記号がオフ・ラインＣＰＵ１２ＢによってⅠ受信される８Ｔ＿ＣｌｋクロックⅠ前にオン・ラインＣＰＵ１２Ａによって受信される。ステップ１０６０及び１０６２は、クロック同期ＦＩＦＯｓを同期するために図４５、４６の説明に関連して上述したのと同じステップを基本的に実行する。オフ・ラインＣＰＵ１２Ａは、ＳＹＮＣ　ＣＬＫ信号で、ＣＰＵｓ及びルータを同期すべく動作する、ＳＬＥＥＰ記号、自己アドレス指定されたメッセージ・パケット、及びＳＹＮＣ記号のシーケンスを送る。一度そのように同期されると、オフ・ラインＣＰＵ１２Ａは、次に、ステップ１０６６で、ス

テップ１０５２でMP１８Aによってセットされた次の状態にルータをまず移動すべく動作する、Ｓｏｆｔ　Ｒｅｓｅｔ（ＳＲＳＴ）指令記号を送る。ルータ１４Ａ，１４Ｂは、いま、オン・ラインＣＰＵ１２Ａへ送られた全てのトラフィックが複製されかつオフ・ラインＣＰＵ１２Ｂへも送られるが、オン・ラインＣＰＵ１２Ａへ送られたものに８クロック遅れるように送られるような、遅延したデュプレックス・モードである。更に、ルータ１４Ａ，１４Ｂは、メッセージ・パケットを受信するのにオン・ラインＣＰＵ１２Ａだけを頼るべくセットされ、かつオフ・ラインＣＰＵ１２Ｂからのあらゆる送信を廃棄する。

【０２３４】ＳＲＳＴ記号は、ＣＰＵｓ１２Ａ，１２Ｂにエコーバックされる（オフ・ラインＣＰＵ１２ＢへのＳＲＳＴ記号が８クロック遅れで）。ＣＰＵｓ１２Ａ，１２Ｂによる受信により、ＣＰＵｓは、ＭＰ１８Aによってセットされた次の状態動作モードへ移動される：シャドウ・モード。手順は、いま、メモリがオン・ラインＣＰＵ１２Ａからオフ・ラインＣＰＵ１２Ｂにコピーされている間に行われるメモリ及び状態（例えば、レジスタ、キャッシュ、等）の監視をセットアップすべくステップ１０８０（図４９）へ移動する。オン・ラインＣＰＵの状態をオフ・ラインＣＰＵにコピーするステップは、オン・ラインＣＰＵの全ての進行中の動作を停止し、全ての構成レジスタ及び制御レジスタ（例えば、インターフェイス装置２４の構成レジスタ７４）、キャッシュ等をオン・ラインＣＰＵのメモリ２８に書込み、メモリ２８の全内容をオフ・ラインＣＰＵにコピーし、かつそれらを一緒に持ち出すリセット・ルーチンに両方のＣＰＵｓを移動する(vectoring) ことによって単に達成されうる。しかしながら、大きなシステムに対して、これは、達成するために数十秒またはそれ以上を費やしうるし、再統一のためにシステム１０をオフ・ラインするのに受容できない時間の量である。この理由により、再統一処理は、オフ・ラインＣＰＵに状態をコピーする動作のほとんどがバックグラウンドで行われる間にオフ・ラインＣＰＵにユーザ・アプリケーション・コードを実行することを継続させる方法で実行される。

【０２３５】しかしながら、オン・ラインＣＰＵがユーザ・アプリケーション・コードを実行し続けるので、どちらかと言えば、オフ・ラインＣＰＵへの状態のコピーの間に、オン・ラインＣＰＵのメモリ２８のセクションは、コピーされた後で変更され、それゆえにコピー・メモリの最初のパスの結果で、オン・ラインＣＰＵメモリは、オフ・ラインＣＰＵのそれに一致しない。これに対する理由は、オン・ラインＣＰＵのプロセッサ２０による通常の処理は、それがオフ・ラインＣＰＵにコピーされた後でメモリ内容を変更することができることである。（オン・ラインＣＰＵ１２Ａのメモリ２８へのＩ／Ｏ書込みは、それらがオフ・ラインＣＰＵ１２Ｂに対してもなされるのでＣＰＵｓ１２Ａ，１２Ｂのメモリの内容を一致させないようにすべくコピー手順に影響を及ぼさない。）再統一の間にこの問題を処理するために二つの機構が用いられる：　まず、再統一処理の間中にメモリ位置がオン・ラインＣＰＵ１２Ａに書込まれたとき、それは、“汚れた(dirty)”とマークされる；第２に、オフ・ラインＣＰＵへのメモリの全てのコピーは、コピーが確認される前に書込まれる（理解されるように、コピーされたメモリ位置を上書きする）ことからオン・ライン・メモリのデータを保護すべくコピーされるメモリ位置を監視する“アトミック・ライト(AtomicWrite)”機構によって実行される。両方の機構は、再統一手順の間だけ用いられ、かつ両方の理解は、メモリ・プリーコピー及びオン・ラインＣＰＵからオフ・ラインＣＰＵへの状態の後続コピーの適切な理解を容易にする。

【０２３６】－　メモリ・マーキング

メモリを“汚れた”とマークすることは、特にこの目的のためにＥＣＣ検査ビットの一つの使用を含む。６４ビット・データ語は、単一ビット誤り訂正、各記憶された語に対する二重ビット誤り検出を供給すべくＥＣＣコードの８ビットで記憶されるということが思い出される。ＥＣＣの８ビットの一つの使用は、１ビット訂正能力（機能）に影響を与えない。しかしながら、それは、２ビット誤りを検出するための能力（機能）を制限しうる。しかし、メモリ複写処理が比較的短い期間の間だけ続くので、このリスクは、受容可能であると信じられる。再統一の間中に、オン・ラインＣＰＵ１２Ａによるメモリ位置への各書込み動作は、その同じメモリ位置の後の(later) 読取り動作に位置を汚れたとマークするように解釈される所定のＥＣＣシンドロームを導き出させるべくＥＣＣ検査コードの１ビットを反転する。（例外がある：オフ・ライン・メモリも同じＩ／Ｏデータで書き込まれるので、第１は、入力Ｉ／Ｏの書込みである。第２は、アトミック・ライト(AtomicWrite) 機構に関連した書込み動作（“書込み条件付き(Write Conditional)”動作(operation) である。）この方法で、オフ・ラインＣＰＵ１２Ｂのメモリ２８によって見られえないオン・ラインＣＰＵ１２Ａのメモリ２８の内容における変化は、マークされ、かつそれらの位置をオフ・ラインＣＰＵ１２Ｂのメモリ２８に続いてコピーすることによって“クリーン(clean)”にされる。

【０２３７】ＥＣＣのどのビットがマーキングに用いられるかは、どのビットが用いられても一致して（矛盾しないで）もちいられるならば、事実まったくそんなに関係ない。

【０２３８】－　アトミック・ライト機構

オフ・ラインＣＰＵへのオン・ラインＣＰＵ１２Ａの状態のコピーは、ルータ１４Ａ，１４Ｂの一つを介してＴＮｅｔ構造を用いているメッセージ・パケット送信の使用を含む。しかしながら、オン・ラインＣＰＵ１２Ａの

メモリ２８へのＩ／Ｏ書込みがオフ・ラインＣＰＵ１２Ｂのメモリへもなされるようにルータ１４Ａ，１４Ｂがデュプレックス・モード動作にセットされるので、オン・ライン状態を含んでいるメッセージ・パケットの転送は、両方のＣＰＵｓ１２Ａ，１２Ｂへ同様に転送されるが、良好な使用にする：オン・ラインＣＰＵ１２Ａによる状態包含メッセージ・パケットの受信は、ルータ１４（１４Ａまたは１４Ｂのどちらか用いられたもの）によるその誤りなし(error-free)受信及び再送信を確認する。また、それは、メモリ位置をクリーンにマークするAtomicWrite （アトミック・ライト）の書込み動作である。それゆえに、コピーされたメモリ位置は、それらが含むデータが、それらをオフ・ラインＣＰＵへ転送し、かつそれが来たメモリへ書込まれる（戻される）メッセージ・パケットに戻し受信されるまでクリーンとマークされない。この方法で、位置は、オフ・ラインＣＰＵのメモリへ成功裏にコピーされるという確認が行われる。

【０２３９】しかしながら、オフ・ラインＣＰＵ１２Ｂへのコピーのためのオン・ラインＣＰＵ１２Ａのメモリ位置の読取りと、オン・ライン・メモリへコピーされたデータの後続の書戻しとの間で、ＣＰＵ１２Ａのメモリ位置への別の書込み動作が行われると想定する（入力Ｉ／Ｏ書込み、またはプロセッサ書込み動作のいずれか）。（オフ・ラインＣＰＵへコピーされた）戻りデータは、それが最初にきた位置へ書込まれ、位置が含んでいる新しい値がなんであろうと上書きし、位置を良好とマークし、かつオン・ラインＣＰＵ１２Ａの進行中の動作に必要でありうるデータを破壊する。この問題を未然に防ぐために、AtomicWrite 機構がファッションされた。AtomicWrite 機構は、オフ・ラインＣＰＵ１２Ｂへコピーされる処理においてメモリ位置を監視するためにオン・ラインＣＰＵのメモリ・コントローラ２６（図４）を用いる。メモリ・コントローラは、オフ・ラインＣＰＵ１２Ｂへコピーするために読取られたが、それらをクリーンとマークすべく戻りデータでまだ書込まれていない、それらのメモリ位置のアドレスを、それらのレジスタにおいて、追跡すべく再統一の間に動作される一組のレジスタ及び制御状態マシンを用いる。コピーされたデータのＣＰＵへの戻りの前にリスト上の位置への（戻されたもの以外の）データの介入書込みが存在すると、リストは、それによってマークされる。コピーされたデータがオン・ラインＣＰＵへ戻されるときには、それがメモリに書込まれる前にリストは、検査される。位置が合間（インターリム）に書込まれたとマークされたならば、戻りデータは、廃棄され、かつメモリ位置が汚れたとマークされたまま残される。他方、オフ・ラインＣＰＵ１２Ｂへのコピーのために読取られているのでメモリ位置が書込まれていないならば、戻りコピー済みデータは、位置へ書戻されかつそれらの位置がクリーンとマークされる。

【０２４０】本質的に、AtomicWrite 機構は、二つの動作を用いる：“ReadLinked（読取りリンクされた）”メモリ動作及び“WriteConditional（書込み条件付き）”動作。ReadLinkedメモリ動作は、オフ・ラインＣＰＵ１２Ｂへコピーされるべきメモリ２８におけるオン・ラインＣＰＵ１２Ａの位置を読取り、ＭＣ２６によって保守されるリンク表におけるその動作のアドレスを保管し、かつそれがオフ・ライン（並びにオン・ライン）ＣＰＵへメッセージ・パケットとしてアセンブリされかつ送られるＢＴＥ８８のキューにコピーされるべきデータを導入すべく機能する。ReadLinked動作のアドレスを保管することは、データをメモリ位置へ戻しかつリンク表のエントリをクリアする、将来のWriteConditional動作にそれを“links （リンクする）”。一般に、動作は、通常のブロック読取りであり、多数のメモリ位置からデータのブロックを生成する。リンク表に書込まれるアドレスは、メモリ位置のブロックのヘッドまたはエンドにおけるメモリ位置のものである。コピーされたメモリ位置のブロックから読取られたデータを含んでいるメッセージ・パケットがオン・ラインＣＰＵ１２Ａによって戻し受信されたときに、それは、WriteConditional動作でメモリ２８へ書込まれる。しかしながら、データが書戻される前に、ＭＣ２６は、リンク表を検査する。ブロック内のメモリ位置が別の動作によって書込まれたならば（例えば、プロセッサ２０による書込み、Ｉ／Ｏ書込み、等）、その以前の書込み動作は、リンク表における位置にフラグを立てる（並びに書込まれたメモリ位置を汚れたとマークする）。ＭＣ２６は、フラグに気付き、かつそれを書込むことなくWriteConditionalデータを廃棄し、汚れたとマークされたメモリ位置をそのまま残し、それらがまだオフ・ラインＣＰＵ１２Ｂへコピーされなければならないことを示す。

【０２４１】再統一処理、及びここで図４９に戻ると、メモリ追跡（AtomicWrite 機構及びメモリ位置をマークすべくＥＣＣを用いて）は、ステップ１０８０及び１０８２でイネーブルされる。これは、再統一（ＲＥＩＮＴ）信号をアサートさせるために再統一レジスタ（図示省略：インターフェイス装置２４の構成レジスタ７４の一つ　－　図9）を書込むことを必要とする。ＲＥＩＮＴ信号は、WriteConditionals 以外の全ての書込み動作、及び全てのＩ／Ｏ書込み動作、に対してＥＣＣ論理回路８５によって生成されたＥＣＣの８ビットの一つを反転するために各メモリ・インターフェイス７０（図１５）のＥＣＣ論理回路８５に結合され、それゆえに、連続的に読取るときに、この反転されたビットを有するデータは、汚れたとマークされているメモリ位置を識別するシンドロームを生成する。そのようにイネーブルされたメモリ追跡で、再統一手順は、オン・ライン・メモリの内容が、下から上へ（または、もし所望ならば上から下へ）、最初のパスでオフ・ラインＣＰＵ１２Ｂのメモ

リへコピーされる（ステップ１０８４）ような、“プリーコピー(pre-copy)”シーケンス（ステップ１０８４－１０８８）へ移る。入力Ｉ／Ｏ及びAtomicWrite 機構以外の書込み動作によって後で書込まれたメモリ位置は、書込まれた位置を汚れたとマークするためにＥＣＣビットを用いる。また、ReadLinked動作によってコピーされた後だが、後続のWriteConditional動作の前の、位置へのメモリ書込みも、マークされる。

【０２４２】メモリ２８の全内容が一度走らされかつオフ・ライン・メモリへコピーされた後、シーケンスは、オフ・ライン・メモリのそれに、即ち、ステップ１０８４の結果で汚れたとマークされたままのメモリ位置に一致しないかもしれないオン・ライン・メモリ位置の増分コピーをここで実行すべくステップ１０８６及び１０８８へ移る。増分コピーは、オン・ライン・メモリ全体を通る多数のパスを含み、合成シンドロームを検査するために各位置を読取る：位置がそれにより汚れたかまたはクリーンかマークされる。汚れたとマークされたならば、位置は、オフ・ラインＣＰＵへコピーされ、かつクリーンとマークされる。位置がクリーンとマークされたならば、そのまま残される。増分コピー動作全体を通して、オン・ライン・プロセッサの通常アクションは、あるメモリ位置を汚れたとマークする。増分コピーの多数のパスは、汚れたメモリ位置がコピーされクリーンにされる割合い（速度）がメモリが汚される割合い（速度）と実質的に等しい点に達するまでステップ２０５２で終了されることが必要である。これを行うために、カウンタは、ReadLinked、WriteConditional、故障したReadLinked、及び故障したWriteConditional動作に対するＭＣ２６に含まれる。メモリを通る各パスの終りでの成功したWriteConditional動作の数を注目することにより、プロセッサ２０は、先のパスと比較した所与のパスの影響を決定することができる。利点が降下したときに、プロセッサ２０は、プリコピー動作を諦める。この地点で再統一処理は、二つのＣＰＵｓ１２Ａ，１２Ｂをロックーステップ動作に置くために準備する。

【０２４３】それゆえに、再統一手順は、ステップ１１００でオン・ラインＣＰＵ１２Ａがフォアグラウンド処理、即ち、ユーザ・アプリケーションの実行を瞬間的に停止する、図５０に示されたステップのシーケンスに移る。次いで、オン・ライン・プロセッサ２０の残りの状態（例えば、構成レジスタ、キャッシュ、等）及びそのキャッシュは、読取られかつメモリ２８のバッファ（メモリ位置のシリーズ）に書込まれる（ステップ１１０２）。次いで、その状態は、両方のＣＰＵｓ１２Ａ，１２Ｂのプロセッサ装置２０をリセット命令に指向する“リセット・ベクトル”と共に、オフ・ラインＣＰＵ１２Ｂにコピーされる。次に、ステップ１１０６は、ＳＬＥＥＰ記号によってルータ１４Ａ，１４Ｂを静止し、ルータのＦＩＦＯｓがクリアであり、プロセッサ・インターフェイス２４のＦＩＦＯｓがクリアであることを確実にすべく自己アドレス指定されたメッセージ・パケットがそれに続き、そして更なる入力Ｉ／Ｏメッセージ・パケットが今後来ない。ステップ１１０８では、オン・ラインＣＰＵ１２Ａは、ＳＲＳＴ記号を両方のＣＰＵｓ１２Ａ，１２Ｂへエコーバックするルータ１４Ａ，１４ＢにＳＲＳＴ指令記号を送信する。エコーバックしているルータは、上述したスレーブ・デュプレックス・モードでまだ動作しているので、オフ・ラインＣＰＵ１２ＢにエコーバックされたＳＲＳＴは、オン・ラインＣＰＵ１２Ａへエコーバックされたその後まだ８クロックである。エコーバックされたＳＲＳＴ記号は、各ＣＰＵのプロセッサ装置２０を、リセット・ベクトルを含むメモリ２８の位置へジャンプさせかつ　プロセッサ装置２０、キャッシュ２２、レジスタ、等へ両方のＣＰＵｓ１２Ａ，１２Ｂの記憶された状態を復元するサブルーチンを起動させるべく、受信されかつ両方のＣＰＵｓ１２Ａ，１２Ｂにより作用される。次に、ＣＰＵｓ１２Ａ，１２Ｂは、同じ命令ストリームを実行し始める。

【０２４４】それゆえに、ステップ１１１２で、ＣＰＵｓ１２Ａ，１２Ｂは、まだシャドウ・モード動作である、即ち、両方は、同じ命令ストリームを実行しているが、ＣＰＵ１２Ｂは、８クロック・サイクルＣＰＵ１２Ａに遅れてそのように行っており、ルータ１４は、ＣＰＵ１２Ｂからの送信を無視すべくまだ構成される。ＣＰＵ１２Ａは、ユーザ・アプリケーションの実行を再開すべくオン・ライン状態へ戻る。再統一手順は、図５１に示したような、“怠惰な再統一(lazy reintegration)”と呼ばれる、再統一の最終ステージにいまここで入る。汚れた位置をマークするためのＥＣＣビットのイネーブリングは、プロセッサが同じメモリに対して同じ事を行っているので、いまここでディスエーブルされなければならない。再統一手順のこのステップの間中、オン・ラインＣＰＵ１２Ａが、命令を実行しているときに（オフ・ラインＣＰＵ１２Ｂも実行している　－　８クロックの遅延にもかかわらず）メモリを読取っているときに汚れたとマークされたメモリ２８の位置に出会うときには、それは、“バス誤り(bus error)”を起動する（ステップ１１２０）。このバス誤りの表示は、同じ命令に対してオフ・ラインＣＰＵ１２Ｂにバス誤りを強要すべく“ソフトーフラグ(soft-flag)”論理回路素子９００（図４４）の選択論理回路９２０を用いて、ＣＰＵ１２Ｂへ送信される（ステップ１１２２）。図４４をちょっと参照すると、ＲＥＩＮＴが、ＭＵＸ９１４を介して、バス誤りがＣＰＵ１２Ａによって出会ったことをＣＰＵ１２Ｂに知らせるためにＣＰＵ１２ＢへのＢＵＳ　ＥＲＲＯＲ信号を選択することをアサートしているということが理解できる。

【０２４５】その間に、ＣＰＵ１２Ａのバス誤りは、（１）誤りの原因及び（２）もし可能ならば誤りの処理

方法を決定すべくプロセッサ装置２０を誤り－処理ルーチンに強要させる。この場合、誤りが、汚れたとマークされたメモリ位置を読取る試みによりもたらされたということが決定される。従って、プロセッサ装置２０は、ＣＰＵ１２Ｂへメモリ位置の内容をコピーすべくAtomic Write 機構を（ＢＴＥ８８を介して－－図９）起動する。次に、ＣＰＵ１２Ａは、バス誤りをもたらした命令を再実行し、かつ進む。ＣＰＵ１２Ａに８クロック・ステップ遅れて動作しているＣＰＵ１２Ｂも、ＣＰＵ１２Ａにバス誤りをもたらしった同じ命令の実行に先駆けて、バス９０２を介してＣＰＵ１２Ａからのその誤りの伝達によって強要されたバス誤りを有する。しかしながら、ＣＰＵ１２Ｂがその命令を実行するときまでには、バス誤りの表示は、ＣＰＵ１２Ｂへ伝達されかつ８クロック後にＣＰＵ１２Ｂの同じ命令と相関される。この相関関係は、オン・ラインＣＰＵ１２Ａからオフ・ラインＣＰＵ１２Ｂへバス誤り信号をパスすることにおける遅延を、ＣＰＵｓへのルータ送信によって導入された８クロック遅延（即ち、シャドウ・モードの８クロック遅延）とマッチングすることによって達成される。しかしながら、ＣＰＵ１２Ｂは、ＣＰＵ１２Ａが起動した同じバス誤り処理ルーチンを通って行くことを強要される。ロックーステップ同期動作に留まるために、オフ・ラインＣＰＵ１２Ｂは、バス誤りルーチン及び“汚れた”メモリ位置からルータへデータを送信することを含んでいる、オン・ラインＣＰＵ１２Ａとまったく同じ動作のシーケンスを実行する。ルータは、ＣＰＵ１２Ｂの送信を無視するが、ＣＰＵ１２Ｂは同じ動作を行うためにＣＰＵ１２Ａによって費やされたのと同じ時間の量を費やさなければならないということに留意する。

【０２４６】その間に、オン・ラインＣＰＵ１２Ａは、ＣＰＵ１２Ａのメモリ全体を通る一つの最後のパスを行うためにある時間を割り当てて、ユーザのアプリケーション・プログラマの実行を続けている間に、まだ汚れたとマークされているそれらのメモリ位置にわたってコピーする。この再統一の最後のステージの間中、メモリ全体は、全てのメモリ位置を検査すべく読取られる。検査されかつ汚れたとマークされるべきであることが見出された全ての位置は、オフ・ラインＣＰＵ、ＣＰＵ１２Ｂへコピーされる。最後に、ＣＰＵｓ１２Ａ，１２Ｂの状態は、二つのＣＰＵｓが真性の、遅延されない、ロックーステップ動作に置かれることができるように一致する。それゆえに、一度、本当に、全てのメモリが検査され、かつ必要ならば、コピーされるということがステップ１１２４で決定されたならば、ステップ１１２８では、ＭＰ１８は、制御論理回路５０９に含まれた構成レジスタへ書込むことによってデュプレックスの次のモード状態にルータ１４Ａをセットする。次に、ＣＰＵ１２Ａは、以前のようにＳＬＥＥＰ、自己アドレス指定されたメッセージ・パケット・シーケンスを発行する。ＣＰＵ１２Ａがルータが静止状態であるということを確実にしたときに、ＣＰＵ１２Ａは、ＳＲＳＴ記号を両方のルータ１４Ａ，１４Ｂへ送る（同時に）。ルータ１４Ａ，１４Ｂによるその記号の受信は、それがＳＲＳＴ記号を二つのＣＰＵｓ１２Ａ，１２Ｂへエコーバックしたときに、それらが両方とも同時にエコーバックされようにそれらをデュプレックス・モードに移動する。ＳＲＳＴ記号がＣＰＵｓ１２Ａ，１２Ｂによっていま受信されるときに、それらは、ＣＰＵｓの両方のプロセッサ装置２０を、同じ仮想時間で同じ状態を有する同じ位置からスタートすべくリセットさせる。ＣＰＵｓ１２Ａ，１２Ｂは、いまロックーステップ動作である。

【０２４７】追加機能

－　低減コスト・デュプレックス・システム

図１～図３をちょっとの間考慮すると、指摘されたように、ＣＰＵｓ１２Ａ，１２Ｂは、別々に、またはデュプレックスされたペアのいずれでも用いうる。前者の場合には、各個別に動作するＣＰＵの設計に用いた冗長は、フェイルーファースト・アーキテクチャを供給する。ＣＰＵｓは、フォルトトレランスに対してソフトウェア・アプローチを実施すべく一つのＣＰＵが“主（１次）”ＣＰＵに、他のＣＰＵが“２次”ＣＰＵに指定されるように（デュプレックスされるのではなく）ペアにされうる。それゆえに、２次ＣＰＵは、主ＣＰＵ上で走っているユーザ・アプリケーションを利用可能であり、かつ主ＣＰＵは、例えば、データ・ベースまたは更新時におけるその地点までの主ＣＰＵの処理の表示である監査ファイルを周期的に更新する。主ＣＰＵが故障したならば、２次ＣＰＵがバックアッパ・アプリケーションを起動しかつデータ・ベースまたは監査ファイルが最後に更新された時点から故障したＣＰＵに取って代わる。これが、ソフトウェア・フォルトトレランス・アプローチである。

【０２４８】ソフトウェア・フォルトトレランス・アプローチは、オペレーティング・システムによって一般に実施される。頑強でなく、従って、この能力（容量）を有していないようなそれらのオペレーティング・システムに対しては、上述したデュプレックスされた動作のモード（図１～図３を参照）が与えられ、二つのＣＰＵｓ１２を用いて同じ命令ストリームの同じ命令を実行すべく動作する。図５２に示したのは、低減コスト・デュプレックスされたＣＰＵｓのペアであり、その一つが他の冗長を有していない。しかしながら、図１～図３を参照すると、ＣＰＵ１２Ａは、両方がデュプレックスされかつロックーステップ・モードで動作しているときに　－

ＣＰＵ１２Ａの個々のプロセッサ装置２０ａ，２０ｂがＣＰＵに対してフェイルーファースト、フォルトトレランスを供給するのと同じ方法で、ＣＰＵ１２Ｂに対する誤り検査冗長で動作することができるということに注目する。それゆえに、図５２に示したように、デュプレ

ックスされた動作に対して、低減コスト・システムが適用可能である。図５２に示したように、処理システム１０’は、上述したように構成されたＣＰＵ１２Ａとルータ１４Ａ，１４Ｂを含む。ここでＣＰＵ１２Ｂ’として示された、ＣＰＵ１２ＡがペアになるＣＰＵは、しかしながら、単一マイクロプロセッサ・ベースＣＰＵシステムとして構成される。また、ルータ１４Ａ，１４ＢとＣＰＵｓとの間の接続は、同じである。

【０２４９】それゆえに、ＣＰＵ１２Ｂ’は、単一プロセッサ装置２０’及び、キャッシュ２２’、インターフェイス装置（ＩＵ）２４’、メモリ・コントローラ２６’、及びメモリ２８’を含んでいる、関連支持コンポーネントだけを備えている。それゆえに、ＣＰＵ１２Ａは、キャッシュ・プロセッサ装置、インターフェイス装置、及びメモリ制御冗長を伴って、図４に示した方法で構成されると同時に、それらのおおよそ半分がＣＰＵ１２Ｂ’を実施するために必要である。動作において、ＣＰＵｓ１２Ａ，１２Ｂ’は、デュプレックス・モードで動作され、それぞれが、同一の命令ストリームの、同じ命令を実質的に同時に実行する。ＣＰＵ１２Ａは、プロセッサ装置２０及びＣＰＵを構成する他の素子の複製を通してフェイルーファースト動作を供給すべく設計される。更に、発散に対してルータ１４Ａ，１４Ｂによってなされたデュプレックス動作及び検査を通して、ＣＰＵ１２Ａは、また、そのコンパニオンＣＰＵ、ＣＰＵ１２Ｂ’に対するチェック・アップを供給する。ペアによって形成された論理ＣＰＵは、発散がルータ１４Ａ，１４Ｂの一つによって検出されかつ発散の検出が故障しているＣＰＵを停止すべく上述したように作用されるされるならば、残りのＣＰＵｓがアプリケーションを続行することができるようなフェイル機能動作を供給する。

【０２５０】残りのＣＰＵが１２Ａであるならば、ＣＰＵ１２Ａを構成する複製されたコンポーネントによる少量のデータ・インテグリティがまだ存在する。生き残っているＣＰＵがＣＰＵ１２Ｂ’であるならば、通常のファッションで実施された誤り検査（即ち、種々のインターフェイスでのパリティ検査）を除き、データ・インテグリティが欠けているということが認識される。図５２は、二つのＣＰＵｓ１２Ａ，１２Ｂ’から出力されたデータの比較を実行すべく一対のルータ１４Ａ，１４Ｂが含まれて、処理システム１０’が示されている。しかしながら、発散検査だけが実行されるべきであるならば一つのルータ１４だけが（例えば、ルータ１４Ａ）用いられるのが必要である。事実、ＣＰＵｓ１２Ａ，１２Ｂ’から出力されたデータを受信すべく接続された二つの入力が、出力されたデータのある程度非同期な受取りを受信すべく上述したようなクロック同期ＦＩＦＯｓを有し、同期ファッションでＦＩＦＯｓから出力したその受信したデータをプリングする、ということを定めて、ルータの使用は、発散に対して必要な検査を実行すべく簡単なコンパレータ回路以外の何者でもないもので置き換えられるということは、当業者には明らかである。

【０２５１】－　スタンバイ・スペアリング(Stanby Sparing)

図１～図３をちょっと参照すると、これらの図に示された処理システムのアーキテクチャの重要な特徴は、各ＣＰＵ１２がそれに利用可能な全てのＩ／ＯＰａｋｅｔ　Ｉｎｔｅｒｆａｃｅ（パケット・インターフェイス）１６のサービス、及びシステムにおける他のＣＰＵ１２の援助（支援）なしで、装着されたＩ／Ｏ装置を有することである。多くの従来の並列処理システムは、特定のプロセッサまたはＣＰＵの援助によってのみＩ／Ｏ装置へのアクセスまたはそのサービスを供給する。そのような場合、Ｉ／Ｏ装置のサービスに対して責任があるプロセッサが故障したならば、Ｉ／Ｏ装置は、システムの残りに対して利用不可能になる。他の従来のシステムは、プロセッサの一つが故障したならば、対応Ｉ／Ｏへのアクセスが残りのＩ／Ｏを通してまだ利用可能であるようにプロセッサのペアを通してＩ／Ｏへのアクセスを供給する。もちろん、両方が故障したならば、再度Ｉ／Ｏは、失われる。また、並列またはマルチ処理システムの他のプロセッサを供給するためにプロセッサの資源を必要とすることは、システムに性能インパクトを課す。

【０２５２】全ての周辺装置へのアクセスをマルチ処理システムの全てのＣＰＵに許容する能力（機能）は、ここでなされたように、上記した米国特許第 4,228,496号公報に教示された“主／バックアップ”処理を拡張すべく動作する。そこでは、多重ＣＰＵシステムは、バックアップ処理が別のＣＰＵｓのバックグラウンドに存在すると同時に、一つのＣＰＵ上で走らせうる主処理を有しうる。周期的に、主処理は、処理の動作に関するデータがバックアップ処理にアクセス可能な位置に記憶されるような“チェックポインティング(check-pointing)”動作を実行する。主処理を走らせているＣＰＵが故障したならば、その故障は、バックアップが存在するものを含んでいる、残りのＣＰＵｓによって検出される。ＣＰＵ故障のその検出は、バックアップ処理を起動させ、かつチェックポイント・データをアクセスするために、バックアップに最後のチェックポイント動作の地点から前の(former)主処理の動作を再開させる。バックアップ処理は、いま主処理になり、残っているＣＰＵｓのプールから、新しい主処理のバックアップ処理を有すべく一つが選ばれる。従って、システムは、最初の故障（即ち、故障したＣＰＵ）が修復される前でさえも別の故障を許容できるような状態に素早く復元される。

【０２５３】それゆえに、処理システム１０の種々の素子を相互接続する方法及び装置は、全てのＣＰＵに、そのシステムの全てのＩ／Ｏ素子、並びにシステムの全てのＣＰＵへのアクセスを供給するということが理解できる。各ＣＰＵは、別のプロセッサのサービスを用いる必

要なしにあらゆるＩ／Ｏをアクセスできる。それにより、システム性能は、Ｉ／Ｏをアクセスすることに含まれる特定のプロセッサを必要とするシステムに比べて向上されかつ改良される。更に、ＣＰＵ１２が故障し、またはラインから取り除かれても、そのアクションは、他のＣＰＵのシステムのＩ／Ｏへのアクセスになんの影響を及ぼさない。

【０２５４】－ トランザクション・シーケンス・プロトコル及びバリア・トランザクション：上述したように、パケットのヘッダ・フィールドは、４ビット・トランザクション・シーケンス番号(Transaction Sequence Number)（ＴＳＮ）フィールドを含む； 図５（ａ）及び図５（ｂ）を参照。ＣＰＵｓ１２ＡまたはあるＩ／Ｏ装置のような、二つ以上の顕著な(outstanding)要求を管理するように構成された処理システム１０の素子（図１～図３）は、ＴＳＮフィールドにおいて各顕著な要求に対して固有のシーケンス番号を供給する。宛先素子が特定の要求に対する応答パケットを生成するときに、応答パケットのＴＳＮフィールドは、応答を促した要求パケットにおけるのと同じＴＳＮ値を含む。次に、応答を受信するシステム素子は、どの要求に応答が対応するかを決定すべく応答におけるＴＳＮにマッチすることができる。ＴＳＮは、応答がもはや存在しない要求に答えるかどうかをシステム素子に決定させる。例えば、ここでのように、あるシステムは、要求に対する応答に所定の期間内に受信されることを要求する。予期したようには応答が受信されなかったならば、要求を起動したシステム素子は、第２の（再度の）要求を単に発行する。早めの要求に対する応答がその後に受信されたならば、システム素子は、応答が答える要求（早めの、無効の、要求、または遅く有効な要求）をＴＳＮから決定することができる。前者ならば、応答は、廃棄される。

【０２５５】ＴＳＮｓは、通称“陳腐なパケット（stale packet）”問題を処理することも支援する。誤りが発生するときに、通過中のメッセージ・パケットは、ネットワークのどこかで動かなくなりうる。これら陳腐なパケットを除去する方法が存在しないならば、それらは、後で現れることができかつシステムが最初の問題から回復した後に動作をもしかすると破壊する。受信した応答メッセージ・パケットのＴＳＮは、受信機に、応答によって運ばれたＴＳＮを応答を促したメッセージ・パケットのＴＳＮと比較することによって応答が現行であるか否かを決定させる。小さなＴＳＮを用いることは、現在顕著である要求にマッチしうるＴＳＮを伴って陳腐な(stale)応答が後で現れるという可能性を生ずる。しかし、大きなＴＳＮフィールドは、伝達されたメッセージ・パケットのそれぞれがより大きいこと、またはそれによってデータ・フィールドが低減されることのいずれかを要求する傾向がある。本発明は、“バリア・トランザクション(Barrier Transaction)”と呼ばれる機構を通してこの問題を解決する。ＴＳＮｓは、用いられるべく継続するが、Barrier Transaction 機構は、ＴＳＮの必要なサイズを ほんの４ビットのフィールドまで低減する。

【０２５６】簡単に言うと、Barrier Transaction は、送信ノードと受信ノードとの間の通信経路のインテグリティを検査するために用いられる。それがＩ／Ｏインターフェイス１６によって発行されうるけれども、Barrier Transaction は、ＣＰＵによって主に起動される。それは、Ｉ／Ｏ装置１７またはＣＰＵ１２向けの先に発行されたメッセージ・パケットに対する予期した応答が所定の割当て期間内に受信されないときに主に用いられる。ＣＰＵ１２は、通常のヘッダ、アドレス、データ、及びＣＲＣフィールドを含んでいる、ＨＡＤＣパケット（図５（ａ））の形のBarrier Transaction メッセージ・パケットを生成しかつ送ることによって経路を検査できる。Barrier Transaction メッセージ・パケットによって運ばれたデータは、トランザクションを独自に識別し、かつそのデータのコピーは、ＣＰＵによる後の比較のためにＣＰＵによって保管される。Barrier Transaction メッセージ・パケットを受信しているシステム素子（例えば、それが別のＣＰＵでもありうるが、Ｉ／Ｏインターフェイス１６の一つ）は、Barrier Transaction 応答を生成しかつ送ることを要求される。しかしながら、そのようにする前に、Barrier Transaction 応答は、それがBarrier Transaction に応答できる前にBarrier Transaction メッセージ・パケットの受信に先駆けて受信した（要求を発行したシステム素子からの）全ての要求を終了または廃棄することを要求される。Barrier Transaction 応答は、Barrier Transaction 要求で運ばれた同じデータを含んでいる、ＨＤＣ形（図６）のものである。Barrier Transaction 応答がトランザクションを起動したＣＰＵによって受信されたときに、応答におけるデータは、応答が対応するBarrier Transaction（多数の顕著なBarrier Transaction が存在しうる）を特に決定すべく、（ＣＰＵによって早めに保管された）早めに送られたBarrier Transaction メッセージ・パケットに存在したデータと比較される。

【０２５７】システム素子と他のシステム素子（例えば、ＣＰＵ１２ＡとＩ／Ｏ17_n ；図１～図３）の間にただ一つの有効経路が存在するので、かつメッセージ・パケットがその宛先へ向かう途中で他のメッセージ・パケットを渡すことができないので、メッセージ・パケット受信のシーケンスは、それらが送られたようなシーケンスにマッチする。それゆえに、Barrier Transaction は、Barrier Transaction を発行しているシステム素子とBarrier Transaction を受信しかつそれに応答しているシステム素子との間の経路をクリアすべく動作する。Barrier Transaction 応答の受信は、全ての要求はBarrier Transaction が答えられる前に送るか、またはやっ

て来ないかのいずれかであることをBarrier Transaction を発行したシステム素子に知らせる。それゆえに、無回答要求が再発行され、応答が最終的に受信されたならばそれは、再発行された要求の結果であり、早めの（及び先に無回答であった）要求に対する遅延応答でないということを知る。Barrier Transaction 機構は、ほんの２～３のＴＳＮ番号の使用を許すということが理解できる。（ここでは、ある程度大きなフィールドを必要としうる従来のシステムに対して、ほんの４ビットのＴＳＮフィールドが用いられる。）

バリア・トランザクションの動作は、カスケード・ルータ１４Ａ及び１４Ｘ、を含むＸ経路によりＩ／Ｏパケット・インターフェイス１６Ａに結合されたＣＰＵ１２Ａと、ＴＮｅｔリンクＬ（即ち、リンクL_x、Ｌ（１）、及びＬ）を示す図５３に示されている。上述したように、各ルータは、エラスティックＦＩＦＯｓ５０６を含むポート入力５０２を有する。この説明に対して、エラスティックＦＩＦＯだけが必要であり、従って示されている。

【０２５８】ルータ１４Ａ及び１４ＸとリンクＬ（１）’との間のリンクＬ（１）のアウトバウンド・セクションは、図５３において破線で示したように、使用不可能になると想定する。これは、多数の理由で発生しうる：故障しているコネクタ、ずれたケーブル、等。ルータ１４Ａからルータ１４Ｘへのアウトバウンド・メッセージ・トラフィックは、中断する（やめる）。Ｉ／Ｏパケット・インターフェイス１６Ａに向かう途中であるが、故障しているリンク・セクションＬ（１）’のまだ上流であるＣＰＵ１２Ａによって起動されたメッセージ・パケット・トランザクションは、応答されず、従って、通信経路における故障を示すことが時間切れになる。割込みが内部的に生成され、かつプロセッサ２０（２０ａ，２０ｂ　－　図４）は、バリア要求（ＢＲ）ルーチンの実行を起動する。そのバリア要求ルーチン（ＢＲ）は、応答の欠如に対して時間切れになる各発行されたトランザクション（メッセージ・パケット）に対する各ＡＶＴエントリ（図１８）の許可フィールド（図１９）におけるＰＥＸビットをクリアすることによって経路をまずディスエーブルする。これは、顕著なトランザクションによって促された応答メッセージ・パケットが後で現れたならば、それは、ＡＶＴエントリがその応答に対してアクセスされかつ検査されたときに阻止されることを確実にする；即ち、リンクにおける故障の理由で失速されたのでなく、それらが最終的に宛先に到達する前に一時的に無くなる、メッセージ・パケット。

【０２５９】ある後の時間に、リンクＬ（１）は、修正され、かつルータ１４Ａのエラスティックバッファ５０６’における１１５２で示されたような、いまでは陳腐な(stale) メッセージ・パケットを解放する。リンクＬ（１）の再制定は、ＣＰＵ１２Ａが　それからＩ／Ｏパケット・インターフェイス１６ＡへのＸ経路がいま動作に戻るという可能性をいま認識しているようにＭＰ１８によってＣＰＵ１２Ａに知らされる。しかしながら、ＣＰＵは、（Ｉ／Ｏパケット・インターフェイス１６Ａに対応している適切なＡＶＴエントリにおけるＰＥＸビットをリセットすることによって）その経路をまだイネーブルできない。理由は、Ｉ／Ｏパケット・インターフェイス１６Ａに、全く異なるメッセージ・パケットとしてそれを誤解させて、それによって応答させるべくその最初の宛先（Ｉ／Ｏパケット・インターフェイス）へフローすべく継続している、エラスティックＦＩＦＯｓ５０６’において１１５２で示されたような、陳腐なトランザクション・メッセージ・パケットの可能性である。この問題を防ぐために、かつＣＰＵ１２ＡによってＸ経路が通常のトラフィックに再び用いられる前に、ＣＰＵ１２Ａで実行しているＢＲサブルーチンは、“Barrier Request （バリア要求）”メッセージ・パケットをＩ／Ｏパケット・インターフェイス装１６Ａに送ることによってBarrier Transaction を起動すべくＢＴＥ論理回路８８（図９及び図２３も参照）を用いる。Barrier Request メッセージ・パケットは、メッセージ・パケットのヘッダ（図５（ａ）及び図５（ｂ）参照）に含まれるソース・フィールドのサブフィールドによってそのように識別される。上記したように、Barrier Request メッセージ・パケットのデータ・フィールドは、その特定のトランザクションに対して固有なデータ値を含む。

【０２６０】Barrier Request メッセージ・パケット（即ち、１１５０）が、Ｉ／Ｏパケット・インターフェイス１６ＡのＸインターフェイス装置１６ａによって受信されたときに、それは、そのデータ・セクションが受信したBarrier Request メッセージ・パケット１１５０に含まれた同じ同一で、固有なデータ値を含む、応答メッセージ・パケットをフォーミュレートする。次に、Ｉ／Ｏパケット・インターフェイス装置１６Ａは、ルータ１４Ｘ，１４Ａを介して、ＣＰＵ１２Ａに応答を戻し送信する。バリア要求メッセージ・パケットへの応答がＣＰＵ１２Ａによって受信されるときに、それは、ＡＶＴ論理回路９０’を通して処理される（図９及び図１６も参照）。バリア応答は、他の型のトランザクション以外のバリア応答を終了させるべくエントリの対応許可フィールドにセットされた“Ｂ”フィールドを有するＡＶＴエントリを用いる。（Barrier Transaction が送られたときに、ＡＶＴエントリは、応答を確認するのに用いるためにＣＰＵによって生成された。）

上述したように、各バリア・トランザクションは、それに応じて送信者に戻されるデータ値を含む。この固有な値は、ＣＰＵ（即ち、ＢＲルーチン）に、送られたデータ値を応答で受信したものと比較させて、応答が異なるバリア・トランザクションの一部でなかったことを確実

にする。一度、バリア応答がそれをＣＰＵ１２Ａに戻させると、陳腐なパケットがこの経路に沿ってＦＩＦＯバッファに残るという可能性がもはや存在しない。また、ＣＰＵ１２Ａは、先にディスエーブルされた経路が通常のトラフィックに対して再びいま用いることができるということを制定した。従って、ＣＰＵ１２Ａは、その経路を用いる全てのＡＶＴエントリにおけるＰＥＸ許可フィールドをセットすることによって経路を再イネーブルする。

【０２６１】本発明の全て及び完全な開示がなされたが、種々の代替及び変更が特許請求の範囲の真の範疇から逸脱することなく本発明の種々の態様に対してなされうるということが当業者に明らかになるであろう。例えば、ある一定の誤りを検出できる８ビット／９ビット・コードの形の指令／データ記号の送信において発生しうる誤りの検出を供給するためのスキームが開示された。概念は、９ビット／１０ビット・コード、または、多重バイト・ワイドのような他の同様なコードへ更に導くことができるということが当業者に明らかであるべきである。更に、ルータ１４は、あらゆる数のポートを有すべく構成されうる；指令／データ・パケット・フォーマットは、（ヘッダ、及び他のフィードにより多くのまたはさらに少ないビットを有して）異なりうる；ルーティング・トポロジーは、ルータ１４を用いて、リング、木、ハイパーキューブ、等として形成することができる。

【図面の簡単な説明】

【図１】本発明の教示に従って構成された処理システムを示す図である。

【図２】図１の処理システムのクラスタまたは配置構成を採り入れている図１の処理システムの代替構成の一つを示す図である。

【図３】図１の処理システムのクラスタまたは配置構成を採り入れている図１の処理システムの代替構成の別の一つを示す図である。

【図４】図１～図３の各サブープロセッサ・システムの一部を形成する中央処理装置（ＣＰＵ）を示す略ブロック図である。

【図５】図５は（ａ）～（ｄ）からなり、図４に示すエリア・ネットワークＩ／Ｏシステムを介して入力／出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの構成を示す図である。

【図６】図４に示すエリア・ネットワークＩ／Ｏシステムを介して入力／出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの一構成を示す図である。

【図７】図４に示すエリア・ネットワークＩ／Ｏシステムを介して入力／出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの他の一構成を示す図である。

【図８】図４に示すエリア・ネットワークＩ／Ｏシステムを介して入力／出力データのような情報を運ぶために用いられる種々のメッセージ・パケットの別の一構成を示す図である。

【図９】プロセッサ及びメモリをＩ／Ｏエリア・ネットワーク・システムとインターフェイスするための図４のＣＰＵｓの一部を形成するインターフェイス装置を示す図である。

【図１０】図９のインターフェイス装置のパケット受信機の部分を示している、ブロック図である。

【図１１】図１０に示すパケット受信機のパケット受信機セクションに用いられるクロック同期ＦＩＦＯ（ＣＳＦＩＦＯ）を示す図である。

【図１２】図１１に示すクロック同期ＦＩＦＯ構造の構成のブロック図である。

【図１３】ＣＰＵの二つのインターフェイス装置からの誤り検査外向き送信に対するクロス接続を示す図である。

【図１４】エンコードされた（８Ｂから９Ｂ）データ／指令記号を示す図である。

【図１５】データがデータ誤り検査のためにメモリコントローラに転送される誤りのクロスチェックのために図９のインターフェイス装置によって用いられる方法及び構造を示す図である。

【図１６】処理システムの他の（ＣＰＵの外部の）コンポーネントに図４のＣＰＵのメモリへの読取り及び／又は書込みアクセスをスクリーンしかつ付与するために用いられるアクセス妥当性検査及び変換（ＡＶＴ）表の実施を表わすブロック図である。

【図１７】ＡＶＴ表エントリをアクセスするために用いられるアドレスの形成を示すブロック図である。

【図１８】通常及び割込み要求に対するＡＶＴ表エントリを示す図である。

【図１９】通常及び割込み要求に対するＡＶＴ表エントリを示す別の図である。

【図２０】通常及び割込み要求に対するＡＶＴ表エントリを示す別の図である。

【図２１】メモリのキューに対する及び図４のＣＰＵのプロセッサ装置に対する割込み要求をポスティングする論理回路を示す図である。

【図２２】キュー・エントリに対するメモリ・アドレスを形成するために用いられる処理を示す図である。

【図２３】プロセッサ装置によって図４のＣＰＵのメモリに形成され、かつ図１～図３に示すエリアＩ／Ｏネットワークを介して送られるべきデータを含んでいる、データ出力構成を示し、かつ図１１、１２のパケット送信機セクションを通るエリアＩ／Ｏネットワークへの送信に対するデータ出力構成をアクセスすべく動作する図９のインターフェイス装置のブロック転送エンジン（ＢＴＥ）装置も示すブロック図である。

【図２４】８チェック・ビットと共に連続偶数アドレス

で二つの同時にアクセスされた３２ビット・ワードを含んでいる、データの７２ビットをメモリからアクセスするために図４のＣＰＵのメモリとそのインターフェイス装置との間で一対のメモリ・コントローラによって一部分形成される７２ビットのデータ・パスの構造を示す図である。

【図２５】オンライン・アクセス・ポート（ＯＬＡＰ）を通るそれへの順次アクセスを示している、図４に示す二つのメモリ・コントローラの一つの略ブロック図である。

【図２６】図４の一対のメモリ・コントローラの状態マシン及び誤り検査のために一つを他のもに対して検査するために用いる技術を簡略形で示す図である。

【図２７】図１～図３に示す処理システムのエリア入力／出力ネットワークに用いるルータ装置を示す略ブロック図である。

【図２８】図２７に示すルータ装置の二つのポート入力の比較を示す図である。

【図２９】図２７に示すルータ装置の６つの入力ポートの一つの構造を示すブロック図である。

【図３０】図２７のルータ装置の入力ポートで受信する指令／データ記号の妥当性を検査するために用いる同期論理回路のブロック図である。

【図３１】図２９に示す入力ポートのターゲット・ポート選択論理回路を示すブロック図である。

【図３２】図３１のターゲット・ポート選択論理回路によって行われるルーティング決定を示している決定チャートである。

【図３３】図３１のターゲット・ポート選択論理回路の一部を形成するアルゴリズム・ルーティング論理回路のブロック図である。

【図３４】図２７に示すルータ装置の６つの出力ポートの一つを示すブロック図である。

【図３５】図１１の一対のＦＩＦＯｓを用いて（各ＣＰＵに対して一つ）、処理システムがロック－ステップ（デュプレックス）モードで動作しているときに同期したファッションで図４のデュプレックス・ペアＣＰＵｓに同じ情報を送信するために用いられる方法を示す図である。

【図３６】そのサブ処理システムの種々の素子を動作するために用いられる複数のクロック信号を開発するための図１～図３のサブ処理システムのそれぞれのクロック生成システムを示している略ブロック図である。

【図３７】互いに一対のサブ処理システムの種々のクロック信号を同期するために対になったサブ処理システムのクロック生成システムを相互接続するために用いられるトポロジーを示す図である。

【図３８】ＦＩＦＯのキューに記号を押し付けかつそれらを引き離すために用いられる二つのクロックがかなり異なるときの情況において図１３または図２９及び図３０のクロック同期ＦＩＦＯを制御するために用いられるＦＩＦＯ一定速度クロック制御論理回路を示す図である。

【図３９】ＦＩＦＯのキューに記号を押し付けかつそれらを引き離すために用いられる二つのクロックがかなり異なるときの情況において図１３または図２９及び図３０のクロック同期ＦＩＦＯを制御するために用いられるＦＩＦＯ一定速度クロック制御論理回路を示す別の図である。

【図４０】図３８及び図３９の一定速度制御論理回路の動作を示すタイミング図である。

【図４１】素子を構成するために図１の（または図２または図３に示す）システムの種々の素子に対する保守プロセッサ（ＭＰ）へのアクセスを供給するために用いられるオンライン・アクセス・ポート（ＯＬＡＰ）の構造を示す図である。

【図４２】キャッシュ・ブロック境界を示している、システム・メモリの部分を示す図である。

【図４３】デュプレックス・モードで動作する対になったサブ処理システムのＣＰＵｓ間の非同期可変を処理するために用いられるソフト・フラグ論理回路を示す図である。

【図４４】デュプレックス・モードで動作する対になったサブ処理システムのＣＰＵｓ間の非同期可変を処理するために用いられるソフト・フラグ論理回路を示す他の図である。

【図４５】互いに情報を受け取る図１の処理システムのＣＰＵｓ及びルータのクロック同期ＦＩＦＯｓをリセットしかつ同期するために用いられるフロー図である。

【図４６】互いに情報を受け取る図１の処理システムのＣＰＵｓ及びルータのクロック同期ＦＩＦＯｓをリセットしかつ同期するために用いられるＳＹＮＣ　ＣＬＫの部分を示す図である。

【図４７】デュプレックス・モードで動作している二つのＣＰＵｓ間の相違（ダイバージェンス）を検出しかつ処理するために用いられる手順を概略的に示している、フロー図である。

【図４８】他のＣＰＵｓが処理システムの動作を計れる程度に停止することなく図１に示す処理システムのＣＰＵｓの一つをロック－ステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す図である。

【図４９】他のＣＰＵｓが処理システムの動作を計れる程度に停止することなく図１に示す処理システムのＣＰＵｓの一つをロック－ステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す他の図である。

【図５０】他のＣＰＵｓが処理システムの動作を計れる程度に停止することなく図１に示す処理システムのＣＰＵｓの一つをロック－ステップ、デュプレックス・モー

ド動作にするために用いられる手順を一般に示す他の図である。

【図５１】他のＣＰＵｓが処理システムの動作を計れる程度に停止することなく図１に示す処理システムのＣＰＵｓの一つをロックーステップ、デュプレックス・モード動作にするために用いられる手順を一般に示す他の図である。

【図５２】本発明の教示を組み込んでいる低減コスト・アーキテクチャを示す図である。

【図５３】図１（または図２、図３）のＣＰＵと入力／出力装置との間の通信経路を検査しかつ検証するバリア・トランザクションの動作を示す図である。

【符号の説明】

１０　データ処理システム

１０Ａ，１０Ｂ　サブープロセッサ・システム

１２Ａ，１２Ｂ　中央処理装置（ＣＰＵ）

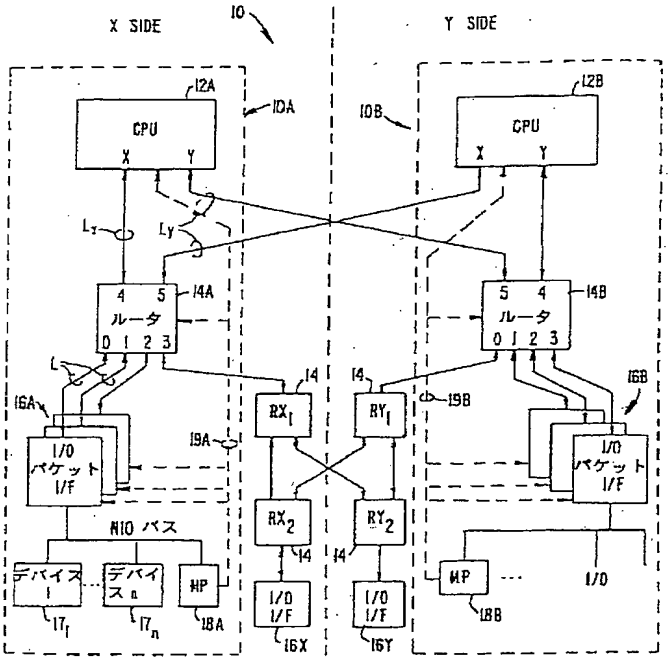
１４，１４Ａ，１４Ｂ　ルータ

１６Ａ，１６Ｂ，１６Ｘ，１６Ｙ　Ｉ／Ｏパケット・インターフェイス

17_1～17_n　Ｉ／Ｏ装置

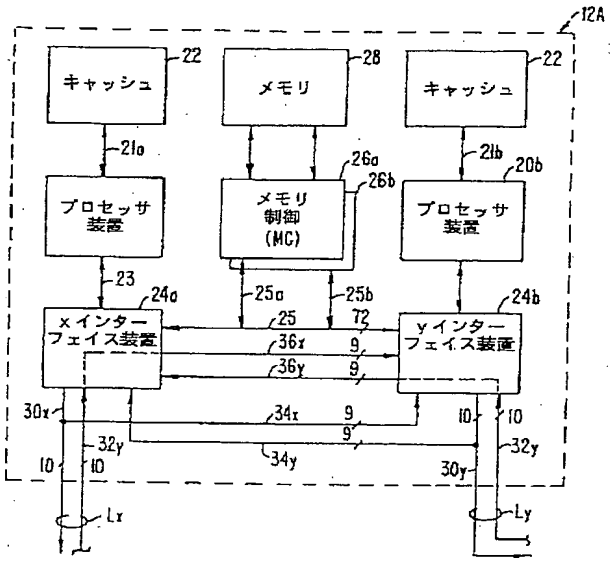
１８Ａ，１８Ｂ　保守プロセッサ（ＭＰ）

【図１】



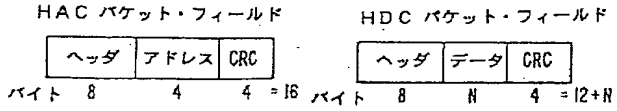
【図２】

【図４】

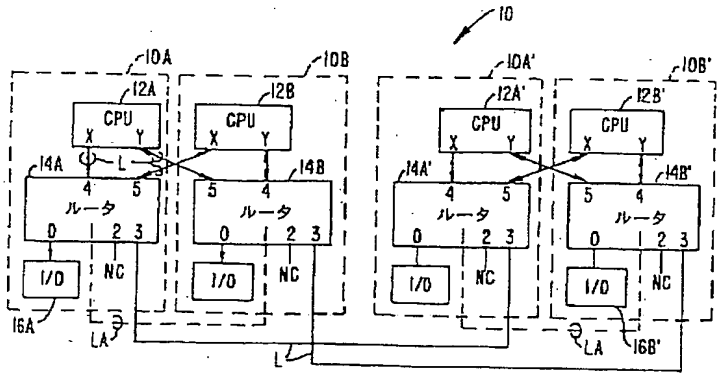


【図６】

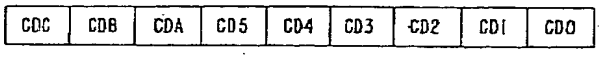
【図７】



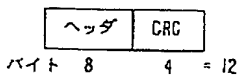
【図８】



【図１４】



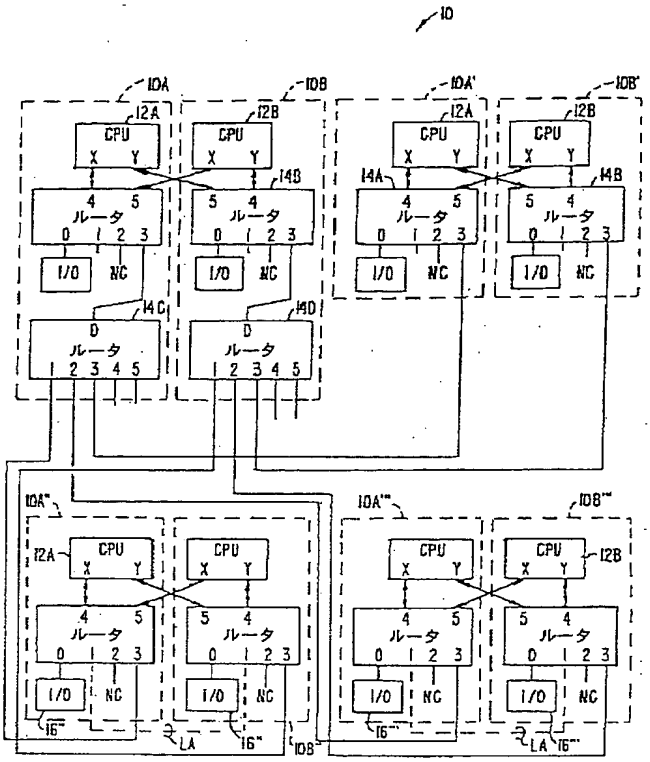
HC パケット・フィールド



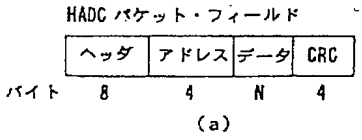
【図１９】

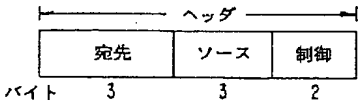
AVT 許可フィールド

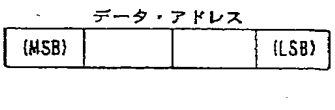
【図３】

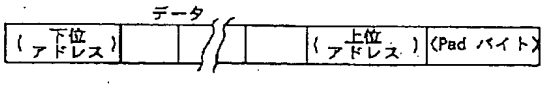


【図５】

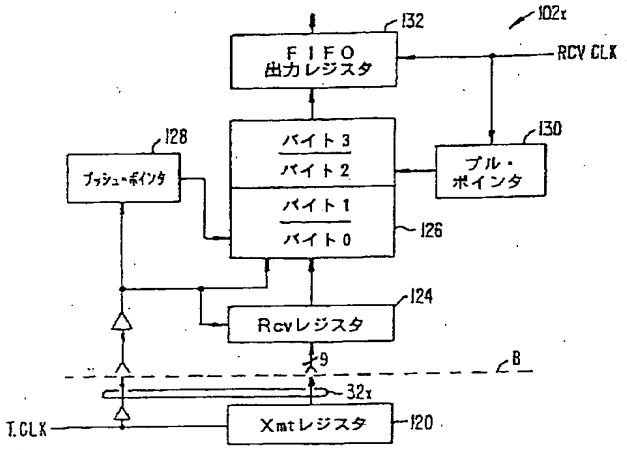




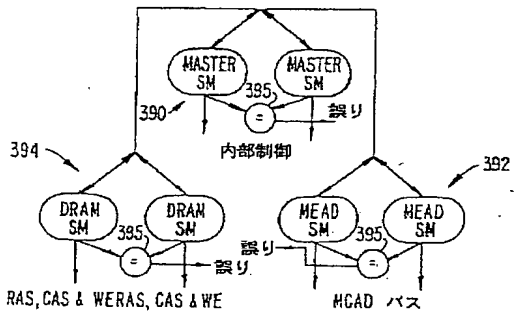




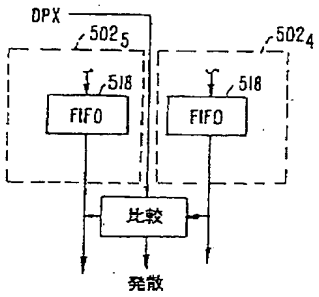
【図１１】



【図２６】



【図２８】

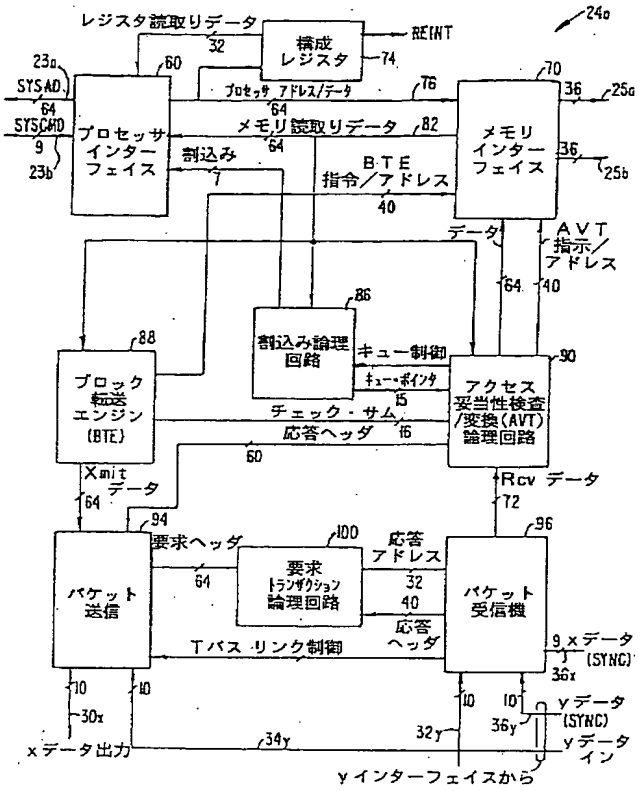


【図１８】

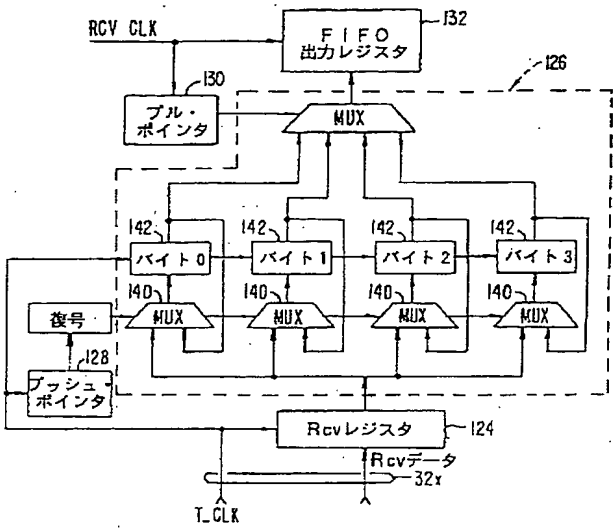
AVT エントリ（標準）

20ビット	52ビット	12ビット	12ビット	20ビット	12ビット
(リザーブド)	物理頁番号	下部バウンド	許可	ソースID	上部バウンド

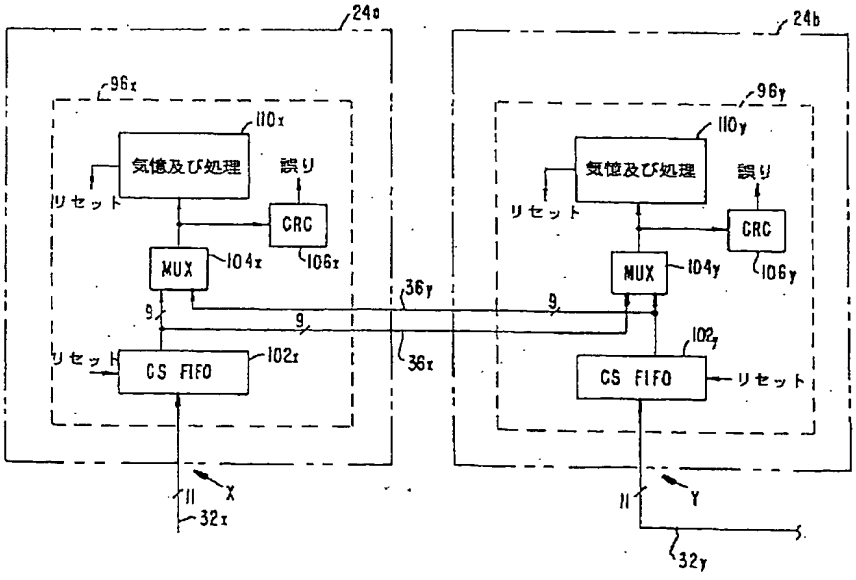
【図９】


レジスタ読取りデータ
構成レジスタ
REINT
SYSAD
SYSCMD
プロセッサインターフェイス
プロセッサ アドレス/データ
メモリ読取りデータ
割込み
BTE 指令/アドレス
メモリインターフェイス
データ
AVT 指示/アドレス
割込み論理回路
キュー制御
キューポインタ
ブロック転送エンジン (BTE)
チェック・サム
応答ヘッダ
アクセス妥当性検査/変換(AVT)論理回路
Xmit データ
Rcv データ
パケット送信
要求ヘッダ
要求トランザクション論理回路
応答アドレス
応答ヘッダ
パケット受信機
Tバス リンク制御
xデータ出力
yインターフェイスから
yデータ (SYNC)
yデータ イン
xデータ (SYNC)

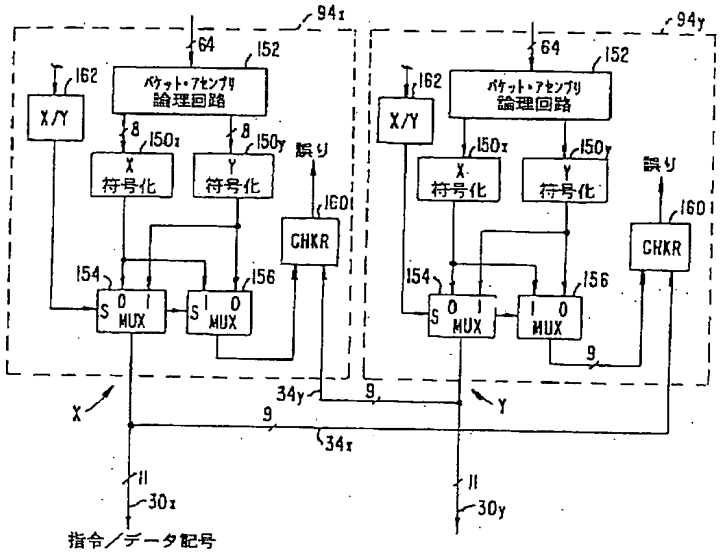
【図１２】


RCV CLK
FIFO 出力レジスタ
プル・ポインタ
MUX
バイト0
バイト1
バイト2
バイト3
復号
プッシュポインタ
Rcvレジスタ
Rcvデータ
T_CLK

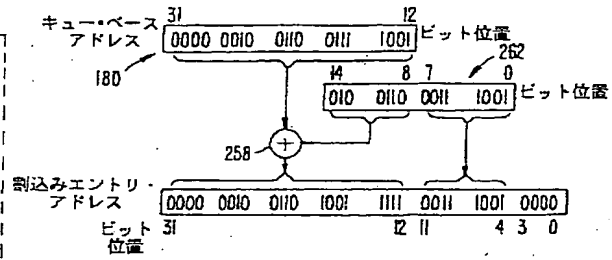
【図１０】


気値及び処理
誤り
リセット
CRC
MUX
CS FIFO

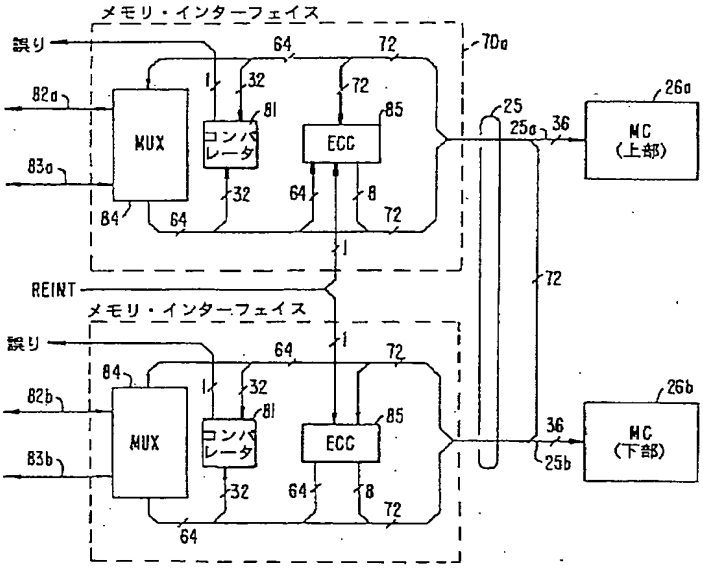
【図１３】



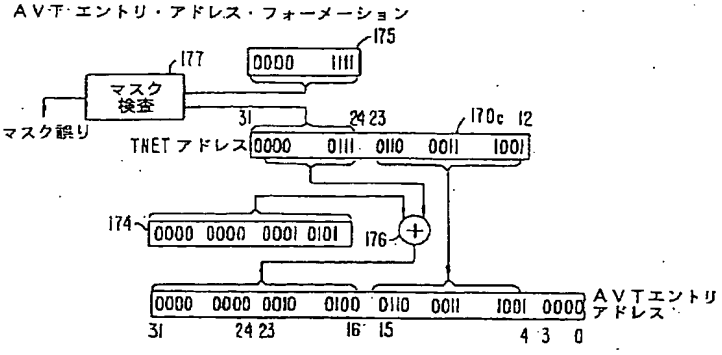
【図２２】



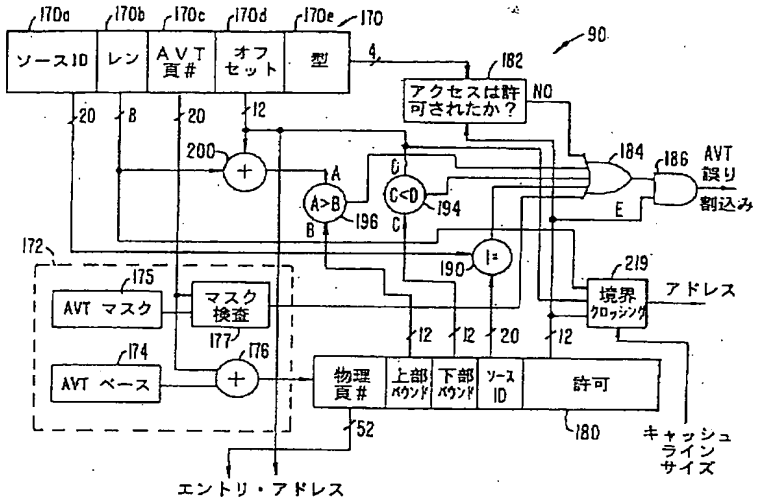
【図１５】



【図１７】



【図１６】

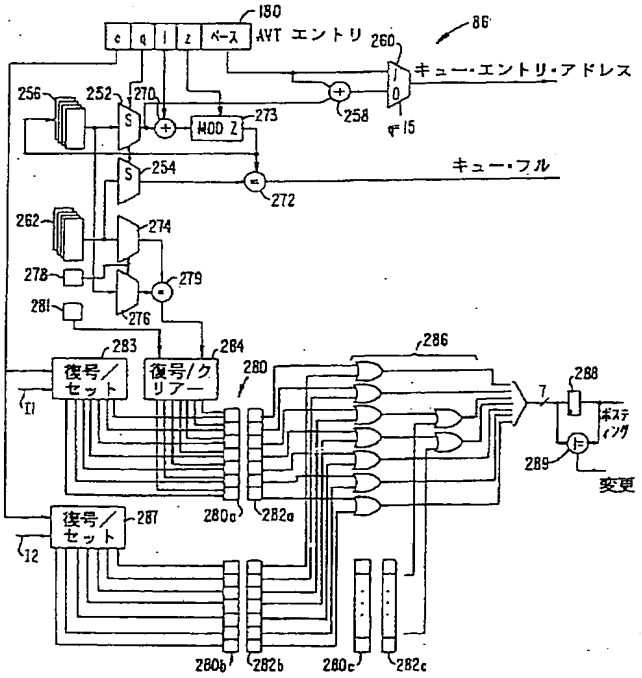


【図２０】

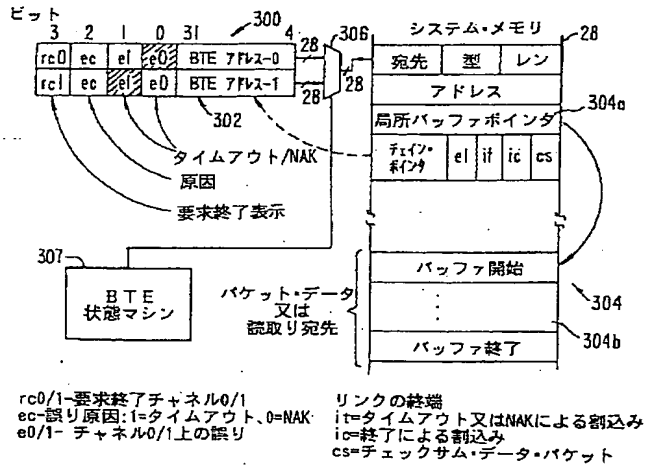
AVT エントリ（割込み）

64ビット	12ビット	20ビット	16ビット	16ビット
キュー・ベースアドレス	許可	ソースID	（リザーブド）	c q I z

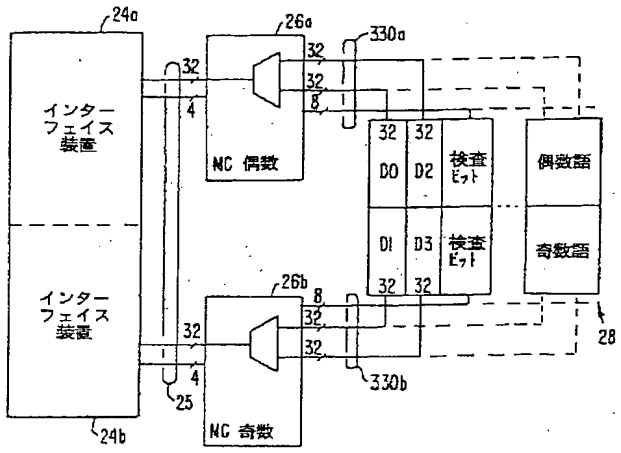
【図２１】



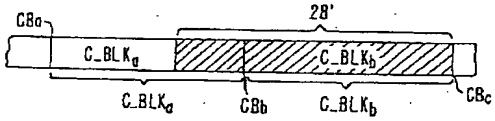
【図２３】



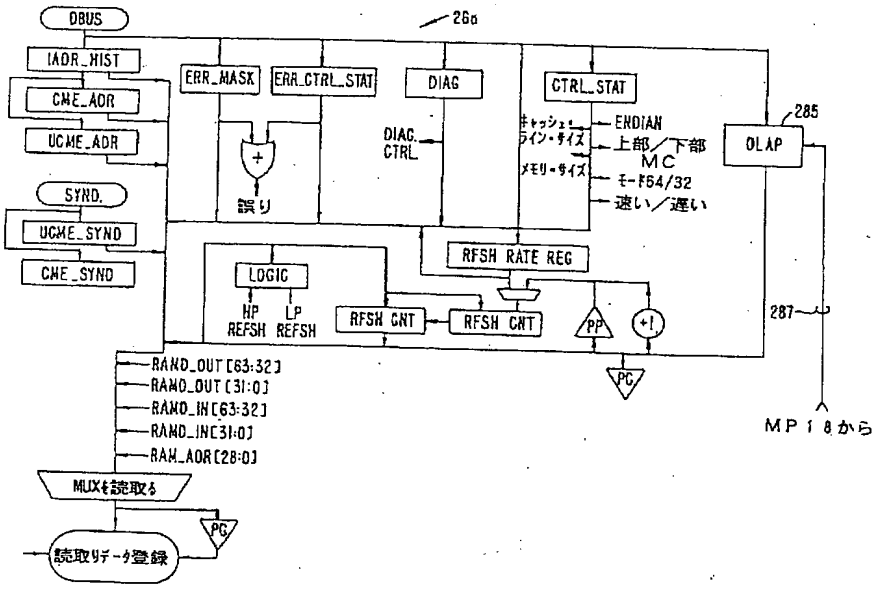
【図２４】



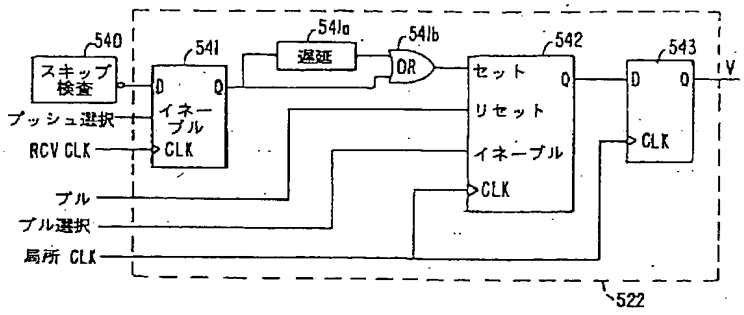
【図４２】



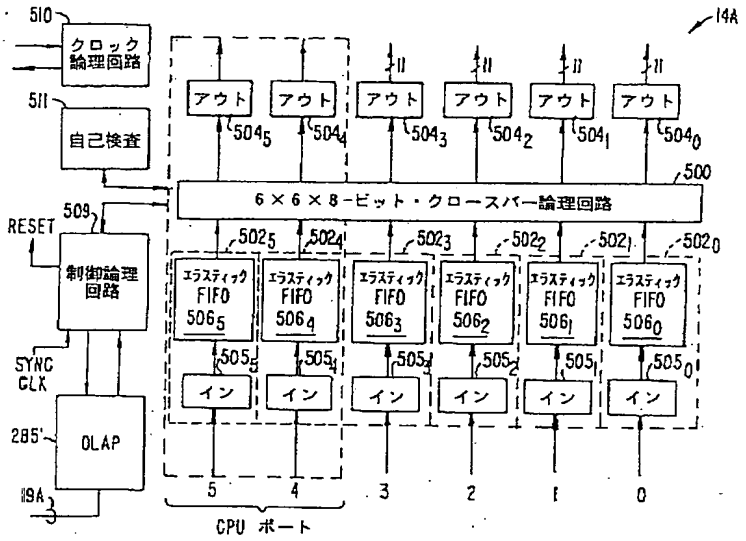
【図２５】



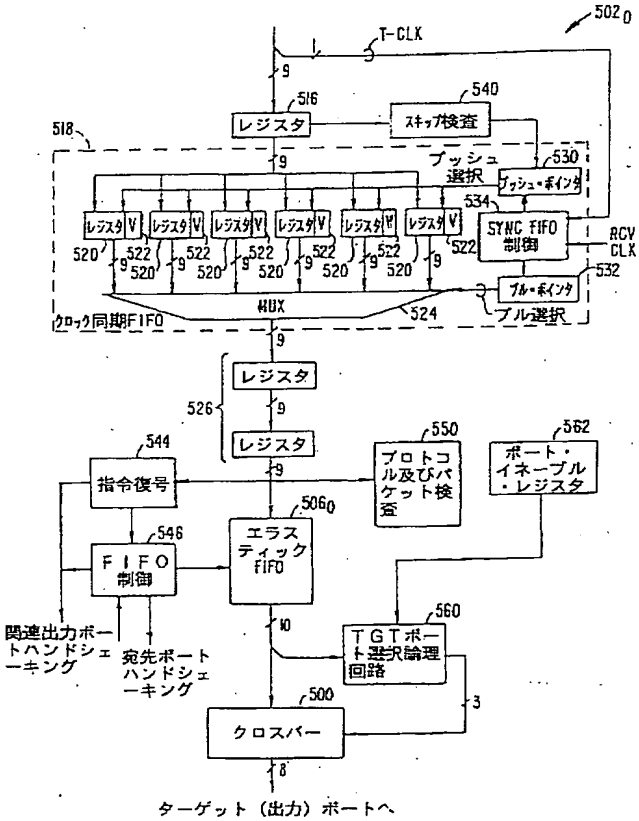
【図３０】



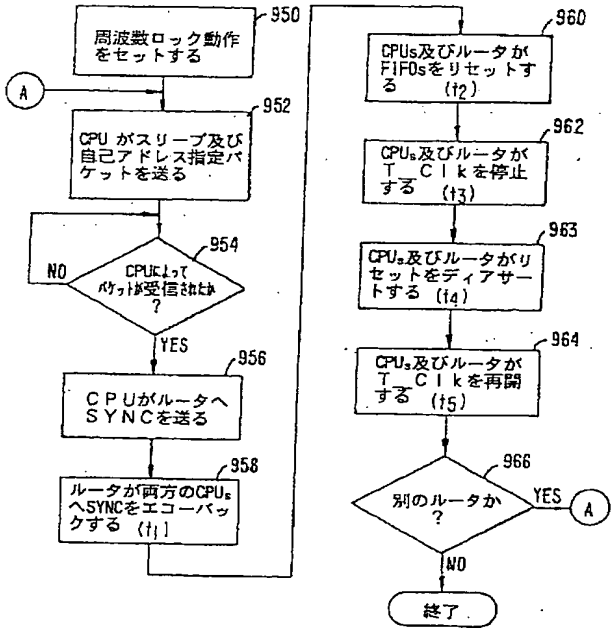
【図２７】



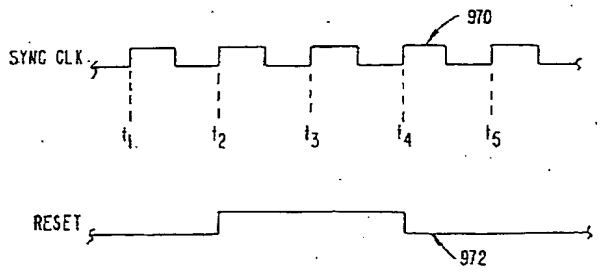
【図２９】



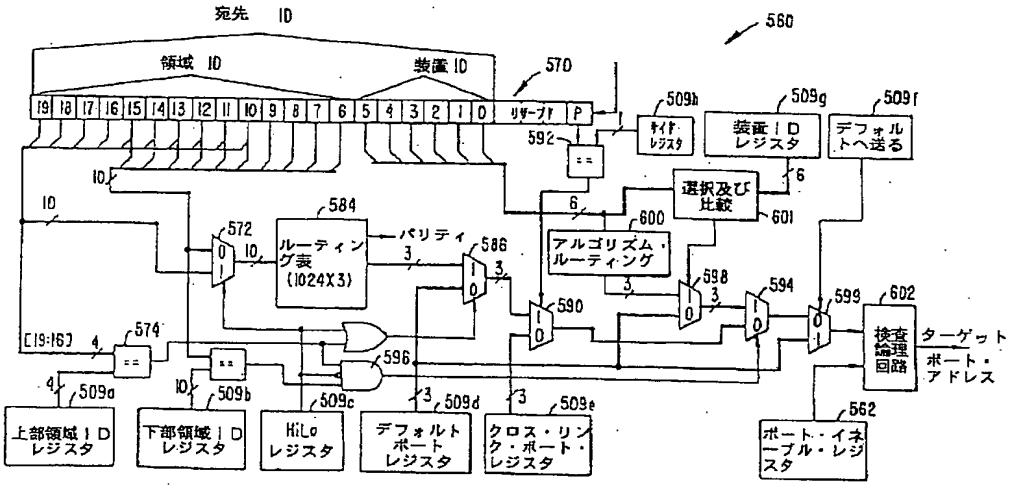
【図４５】



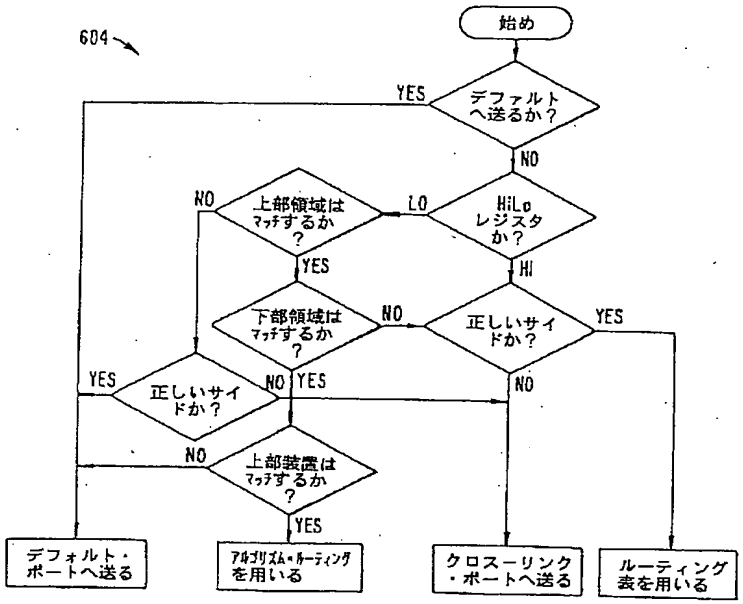
【図４６】



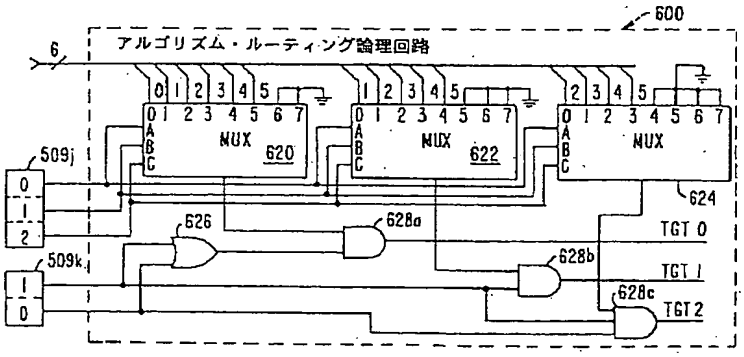
【図３１】



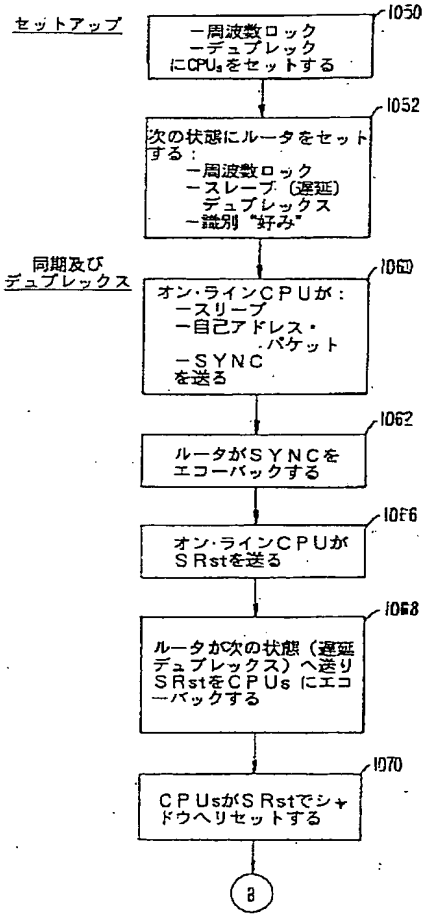
【図３２】



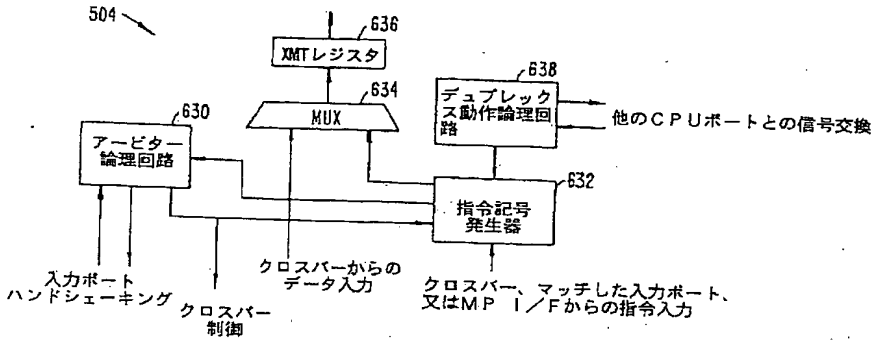
【図３３】



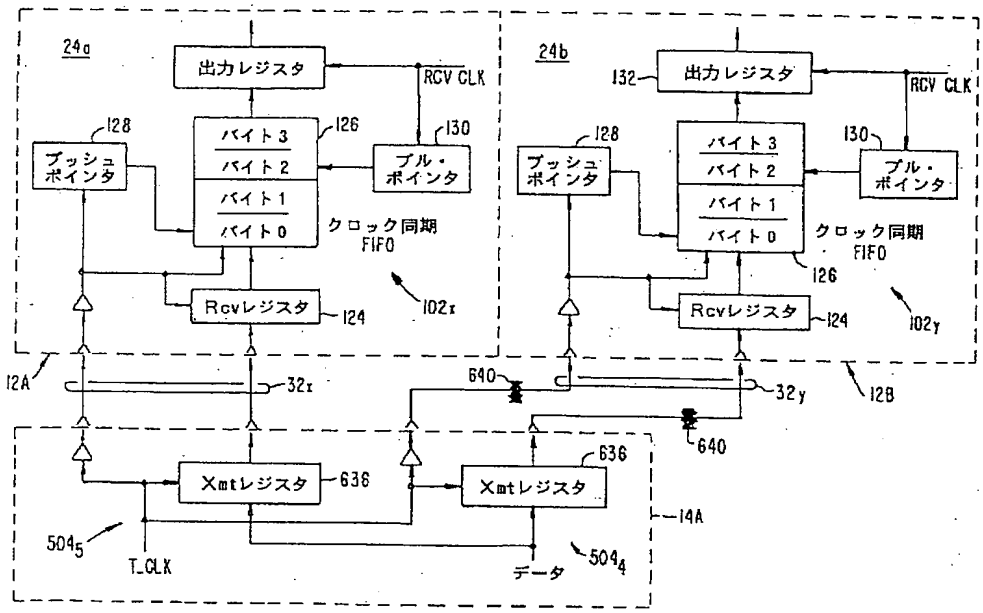
【図４８】



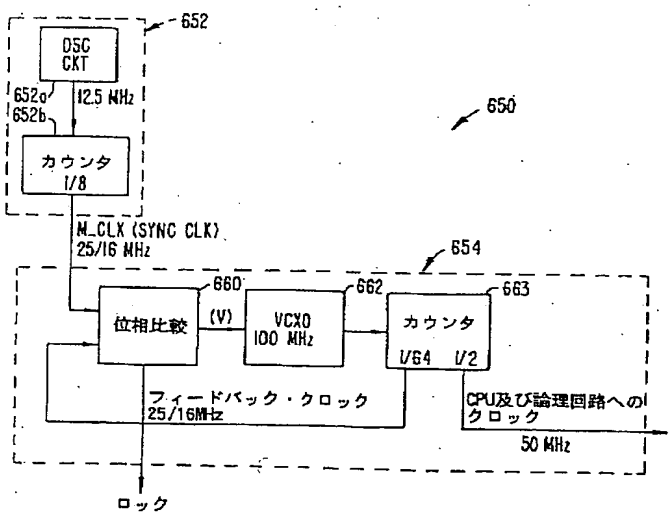
【図３４】


504
636
XMTレジスタ
634
MUX
638
デュプレックス動作論理回路
他のＣＰＵポートとの信号交換
630
アービター論理回路
632
指令記号発生器
入力ポート
ハンドシェーキング
クロスバー制御
クロスバーからのデータ入力
クロスバー、マッチした入力ポート、又はＭＰ　Ｉ／Ｆからの指令入力

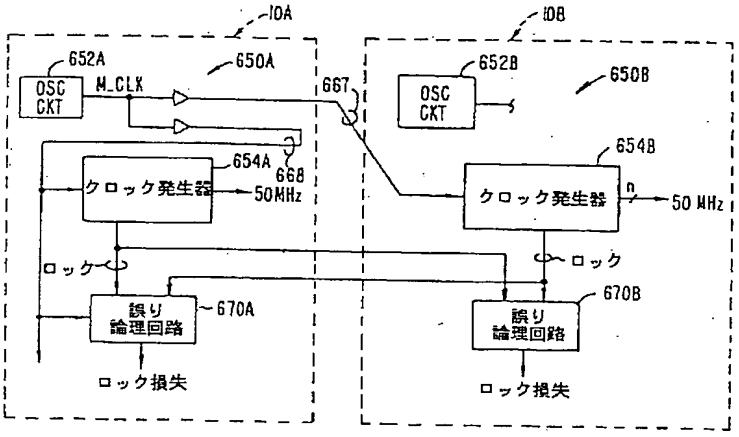
【図３５】


24a
出力レジスタ
RCV CLK
128
プッシュポインタ
バイト３
バイト２
バイト１
バイト０
126
130
プル・ポインタ
クロック同期FIFO
Rcvレジスタ
124
102x
24b
132
出力レジスタ
RCV CLK
128
プッシュポインタ
バイト３
バイト２
バイト１
バイト０
130
プル・ポインタ
クロック同期FIFO
126
Rcvレジスタ
124
102y
12A
32x
640
32y
128
640
636
Xmtレジスタ
636
Xmtレジスタ
14A
504５
T_CLK
データ
504４

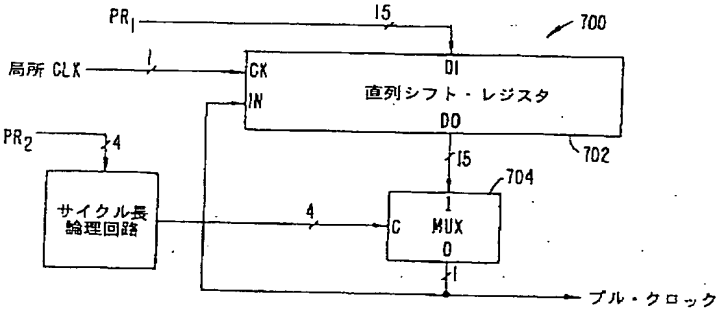
【図３６】


652
OSC CKT
652a
652b
12.5 MHz
カウンタ 1/8
650
M_CLK (SYNC CLK) 25/16 MHz
654
660
位相比較
(V)
662
VCXO 100 MHz
663
カウンタ 1/64 1/2
フィードバック・クロック 25/16MHz
CPU及び論理回路へのクロック
50 MHz
ロック

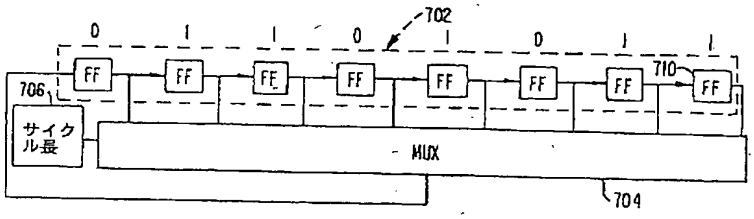
【図３７】



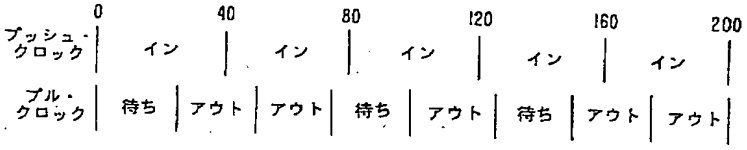
【図３８】



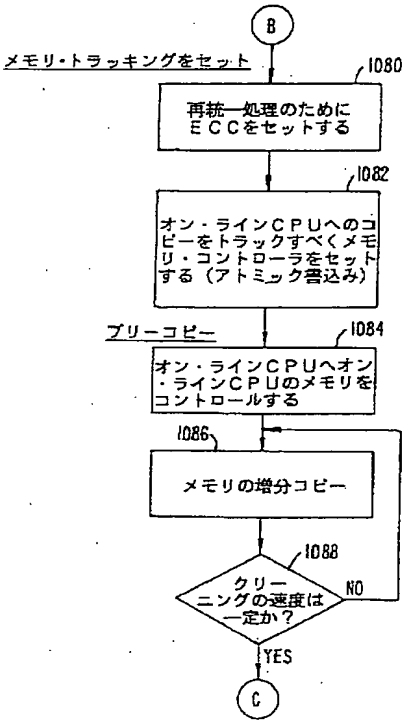
【図３９】



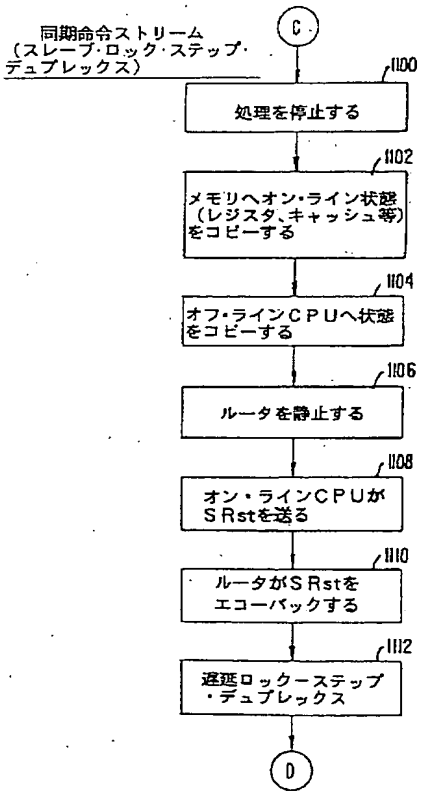
【図４０】



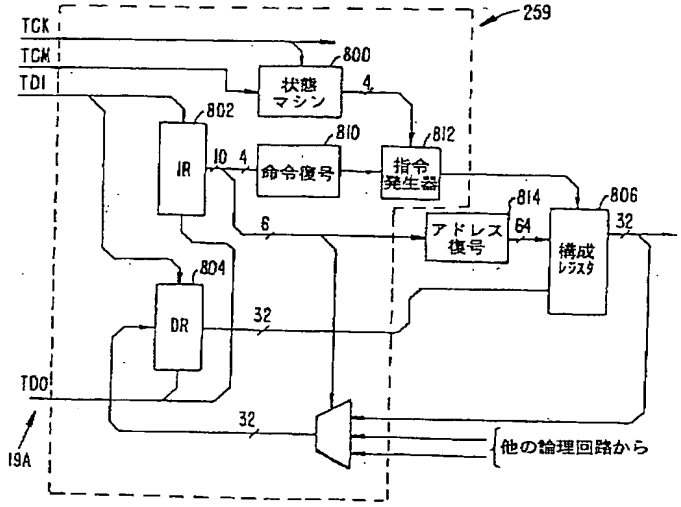
【図４９】



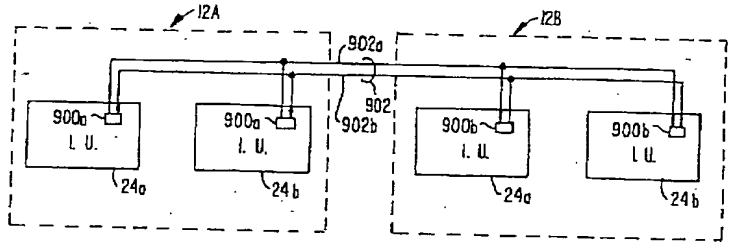
【図５０】



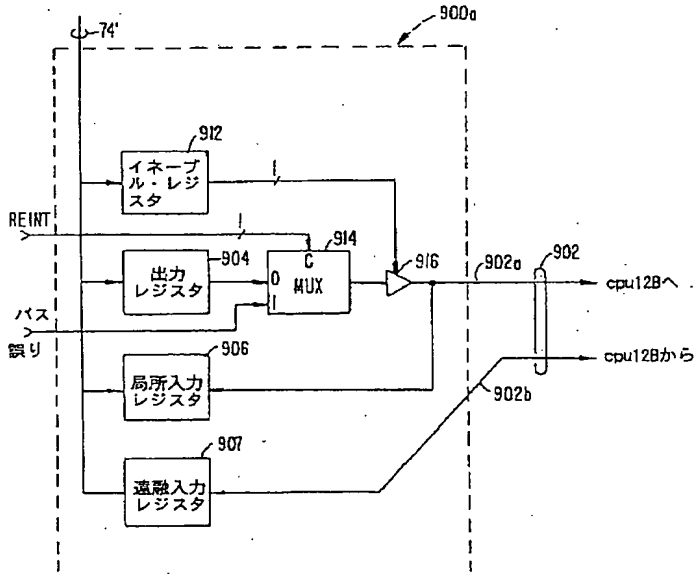
【図４１】



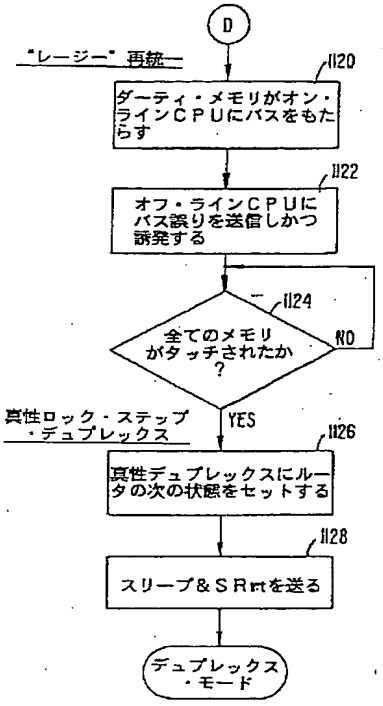
【図４３】



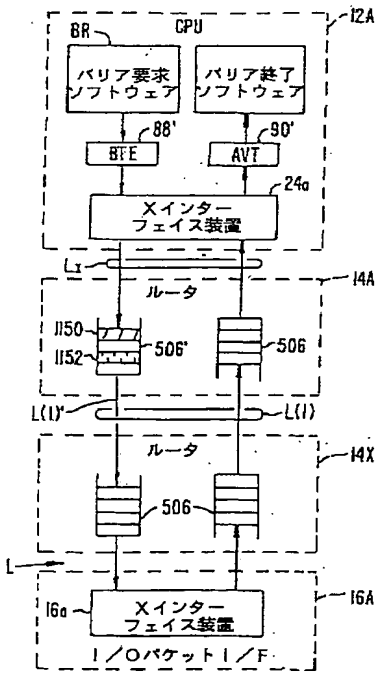
【図４４】



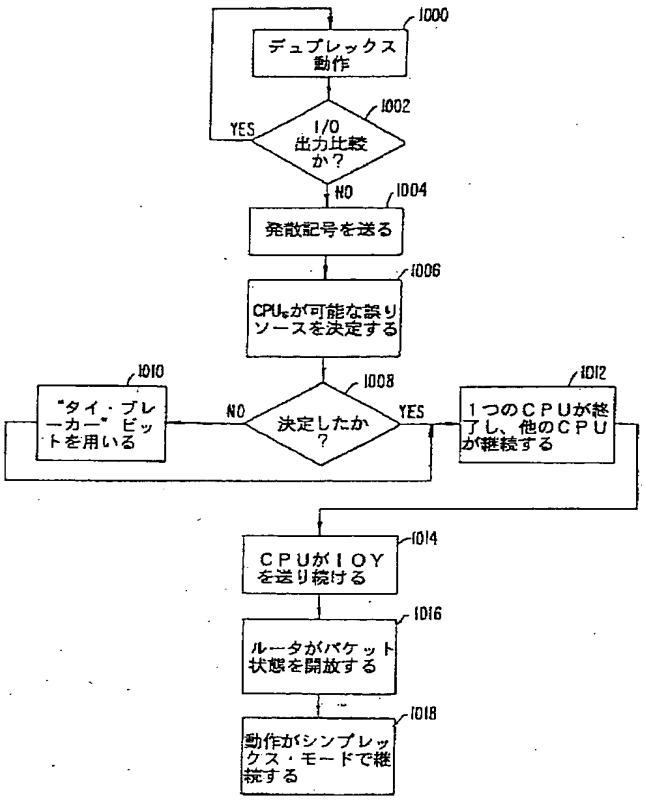
【図５１】



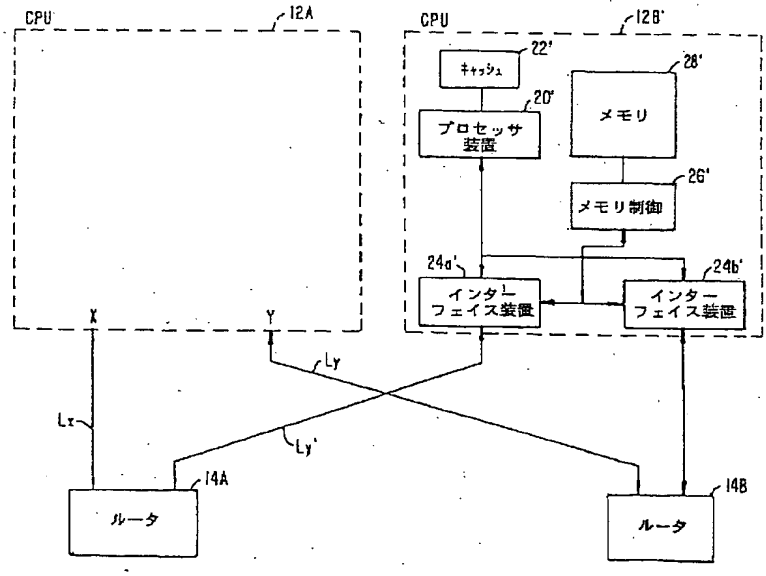
【図５３】



【図４７】


1000
デュプレックス動作
1002
I/O出力比較か？
YES
NO
1004
発散記号を送る
1006
CPUₙが可能な誤りソースを決定する
1008
決定したか？
NO
YES
1010
"タイ・ブレーカー" ビットを用いる
1012
1つのCPUが終了し、他のCPUが継続する
1014
CPUがIOYを送り続ける
1016
ルータがバケット状態を開放する
1018
動作がシンプレックス・モードで継続する

【図５２】


CPU
12A
CPU
12B'
22'
キャッシュ
20'
プロセッサ装置
28'
メモリ
26'
メモリ制御
24a'
インターフェイス装置
24b'
インターフェイス装置
X
Y
Lx
Ly
Ly'
14A
ルータ
14B
ルータ

フロントページの続き

(72)発明者　ディヴィッド　ジェイ　ガーシア
アメリカ合衆国　カリフォルニア州
95030　ロス　ガトス　ハッチンソン　ロード　24100

【公報種別】特許法第１７条の２の規定による補正の掲載
【部門区分】第６部門第３区分
【発行日】平成１５年９月１０日〔２００３．９．１０〕

【公開番号】特開平９－１２８３４９
【公開日】平成９年５月１６日（１９９７．５．１６）
【年通号数】公開特許公報９－１２８４
【出願番号】特願平８－１４５５５０
【国際特許分類第７版】
G06F 15/16 360
470
11/18 310
【ＦＩ】
G06F 15/16 360 R
470 J
11/18 310 A

【手続補正書】
【提出日】平成１５年６月６日（２００３．６．６）
【手続補正１】
【補正対象書類名】明細書
【補正対象項目名】特許請求の範囲
【補正方法】変更
【補正内容】
【特許請求の範囲】
【請求項１】　各々が複数のプロセッサを備え、Ｎが３以上の整数を表すＮ個の中央処理装置と、
複数の入力／出力装置と、
前記Ｎ個の中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの別々でかつ独立した通信アクセスを有するように前記Ｎ個の中央処理装置の各々と前記入力／出力装置とを相互接続するネットワークとを備え、
前記ネットワークは、前記Ｎ個の中央処理装置間でプロセッサ間通信を供給し、前記ネットワークは、前記中央処理装置の前記一つの内でロックステップ動作を可能にすべく前記中央処理装置の一つに対して少なくとも一つが通信を同期する、複数のルーティング装置を含み、
それによって、前記入力／出力装置の一つを採り入れている前記Ｎ個の中央処理装置の故障したものの動作タスクは、前記入力／出力装置の前記一つの採り入れを含んでいるその他のＮ個の中央処理装置のいずれかによって実行されることを特徴とする多重処理システム。
【請求項２】　第１の中央処理装置及び第２の中央処理装置を含んでおりかつＮが２以上の整数である、各々が複数のプロセッサを備え、かつ各々が一つまたはそれ以上の処理を実行すべく動作する、Ｎ個の中央処理装置；前記第１の中央処理装置で実行される主要処理；前記第２の中央処理装置に関連付けられ、前記主要処理に動作及び機能において実質的に同一である、バックアップ処理；複数の入力／出力装置；前記Ｎ個の中央処理装置のいずれか一つが前記入力／出力装置のいずれか一つへの別々でかつ独立した通信アクセスを有するように前記Ｎ個の中央処理装置の各々と前記入力／出力装置とを相互接続するネットワークを備え、
前記ネットワークは、前記Ｎ個の中央処理装置間でプロセッサ間通信を供給し、前記ネットワークは、前記中央処理装置の前記一つの内でロックステップ動作を可能にすべく前記中央処理装置の一つに対して少なくとも一つが通信を同期する、複数のルーティング装置を含むことを特徴とする多重処理システム。